

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

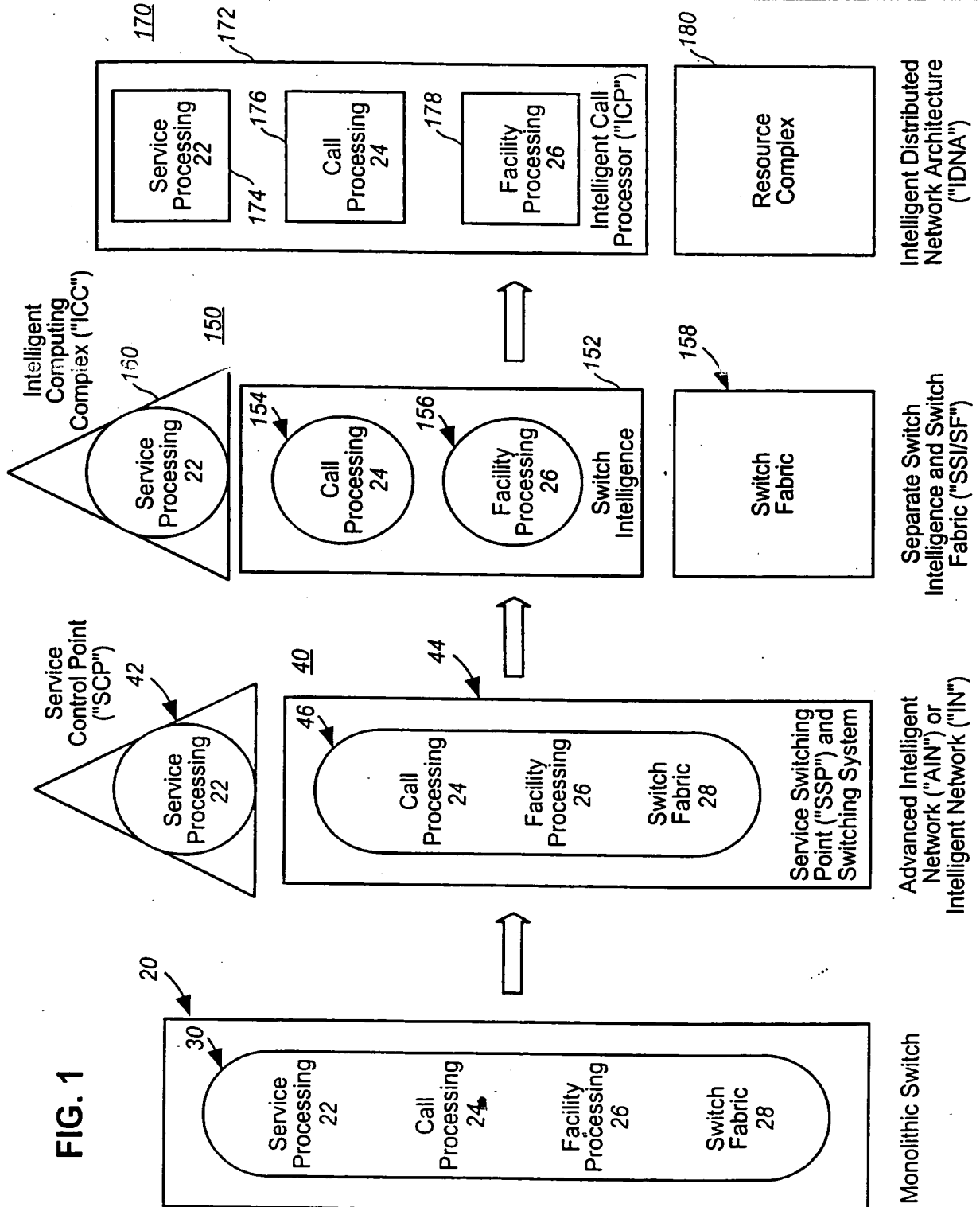
Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

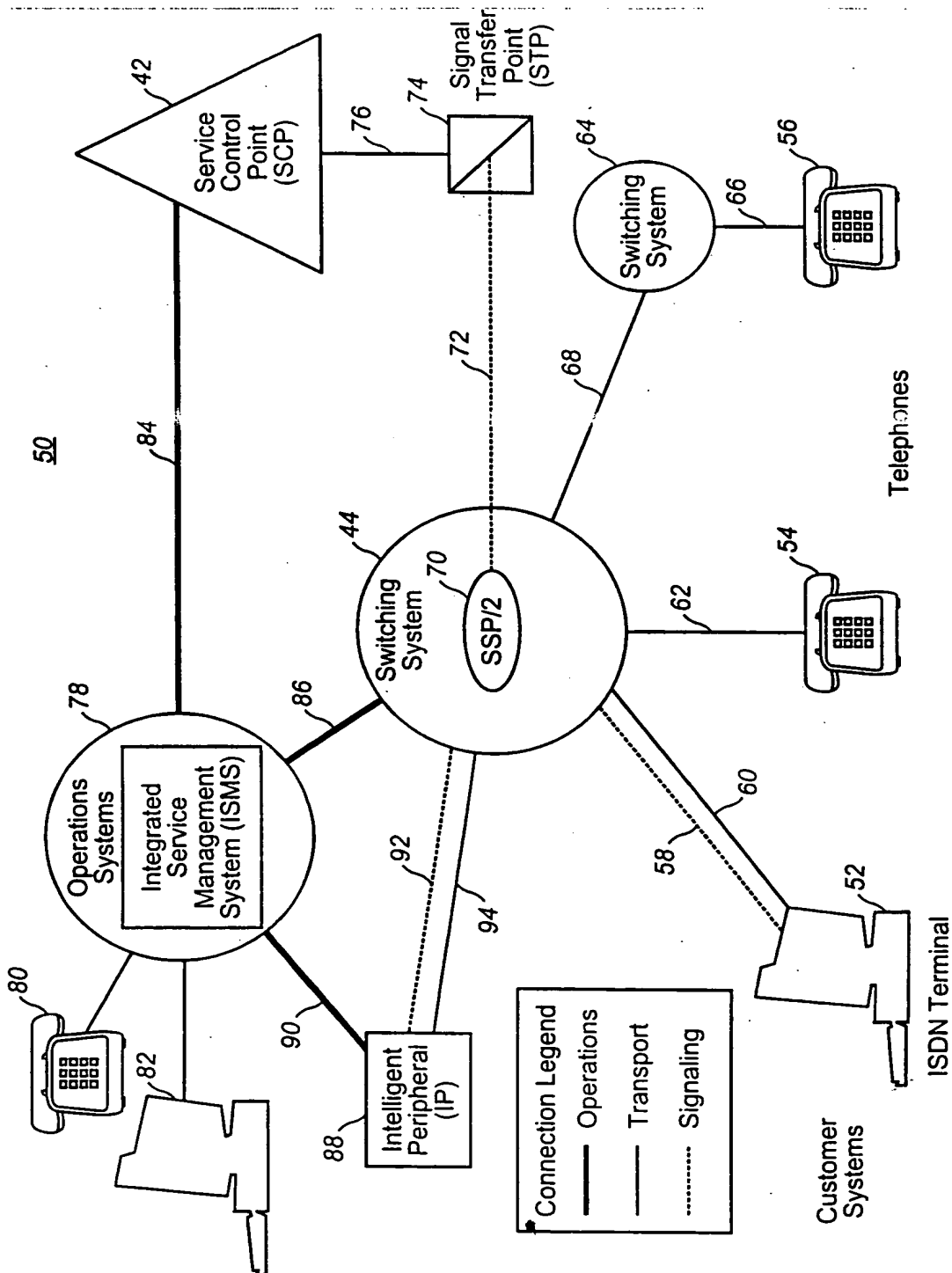
IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

1/61



2/61



3/61

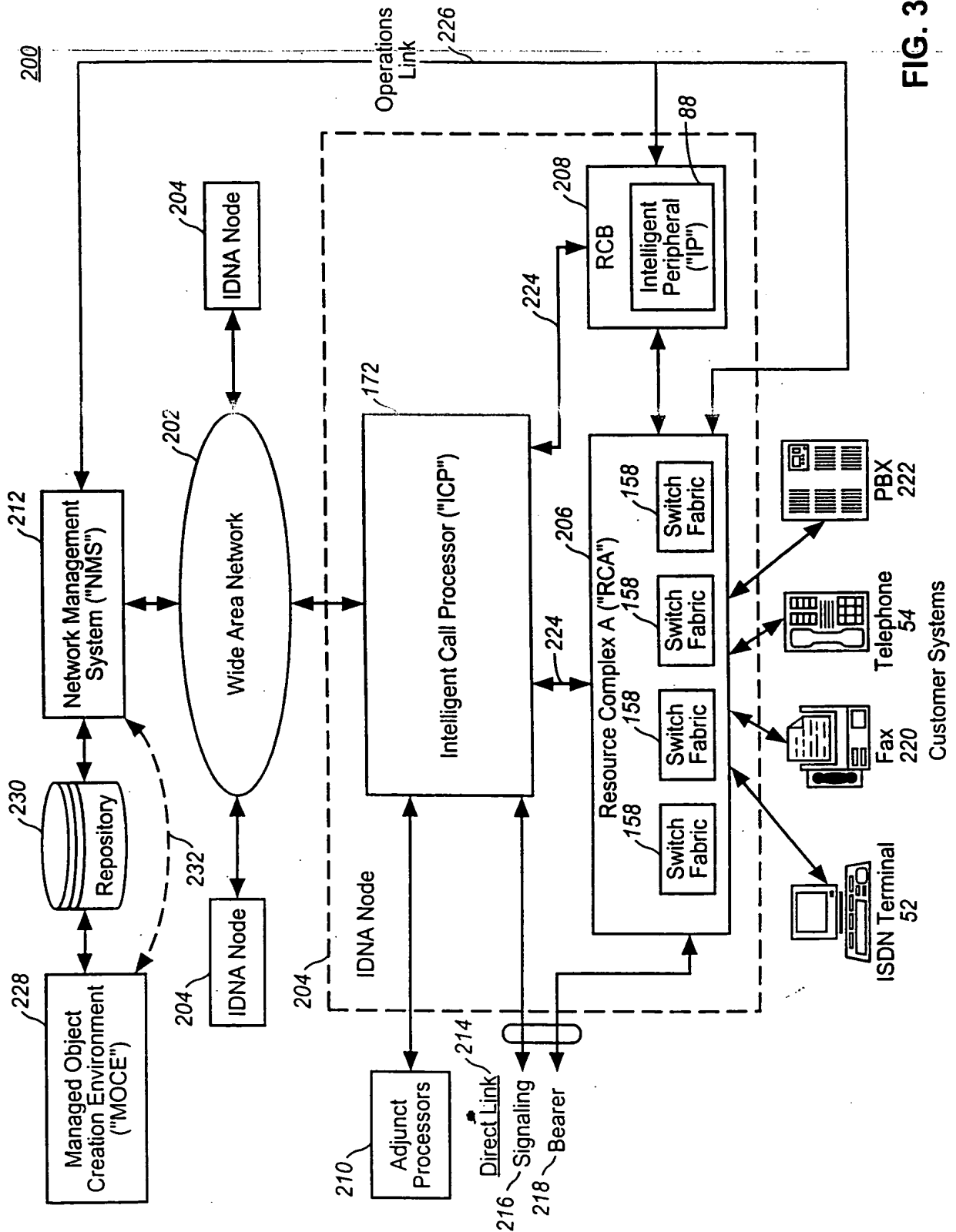


FIG. 3

4/61

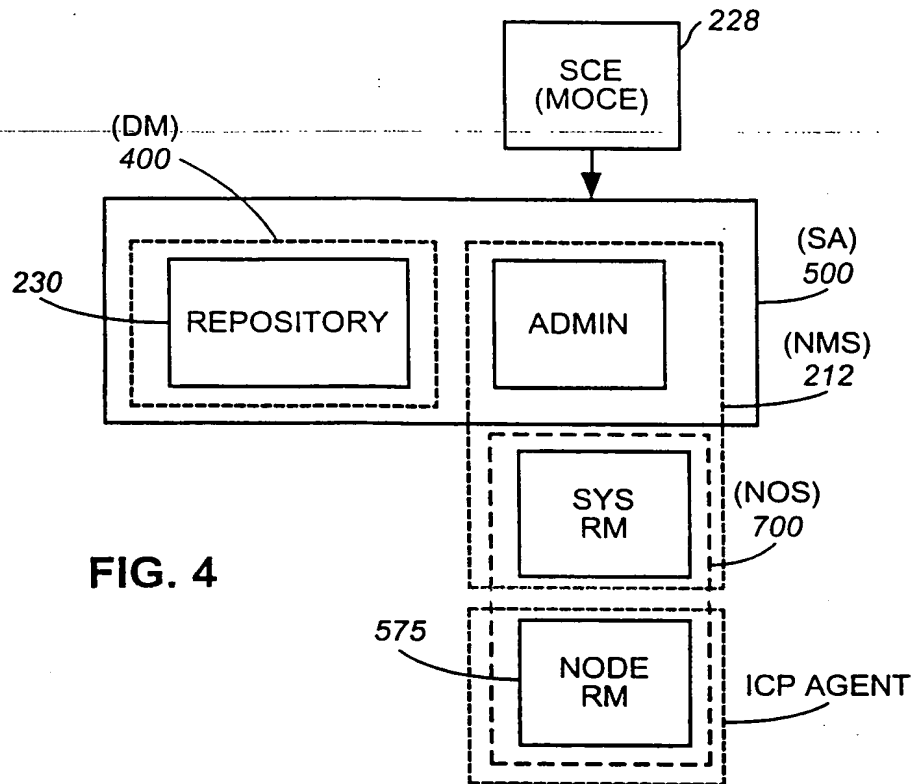


FIG. 4

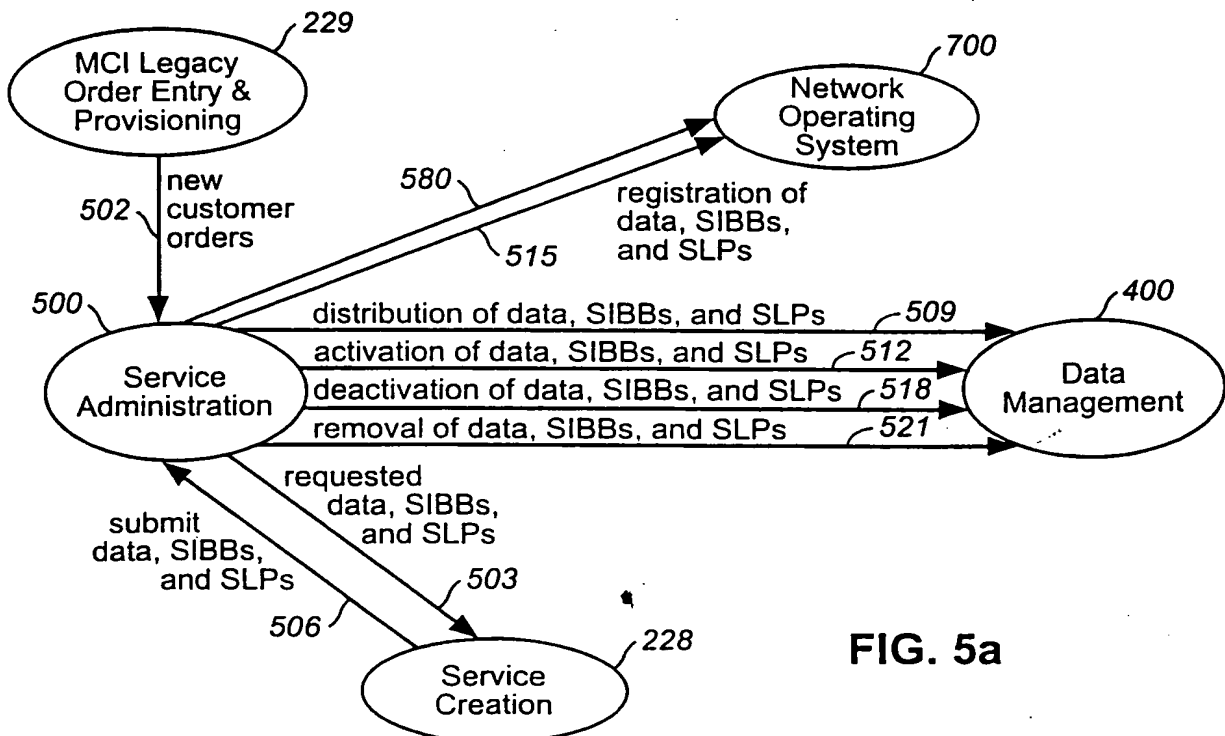


FIG. 5a

5/61

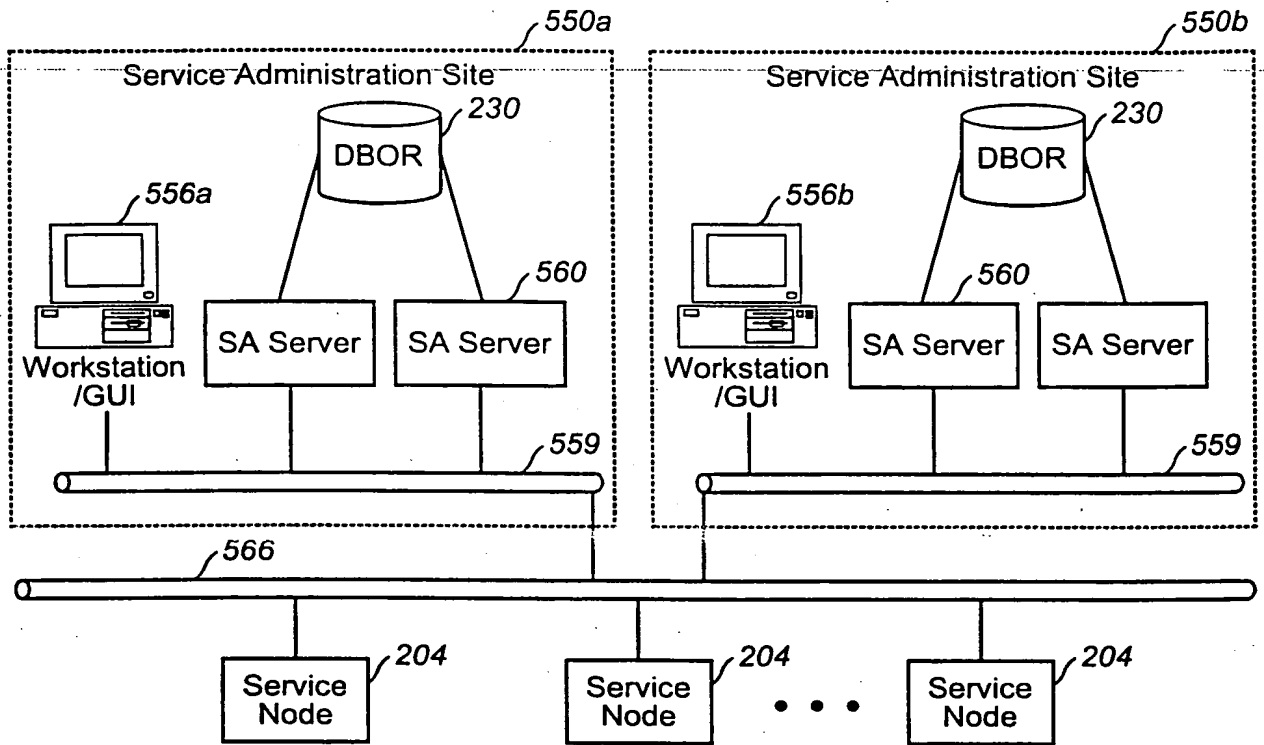


FIG. 5b

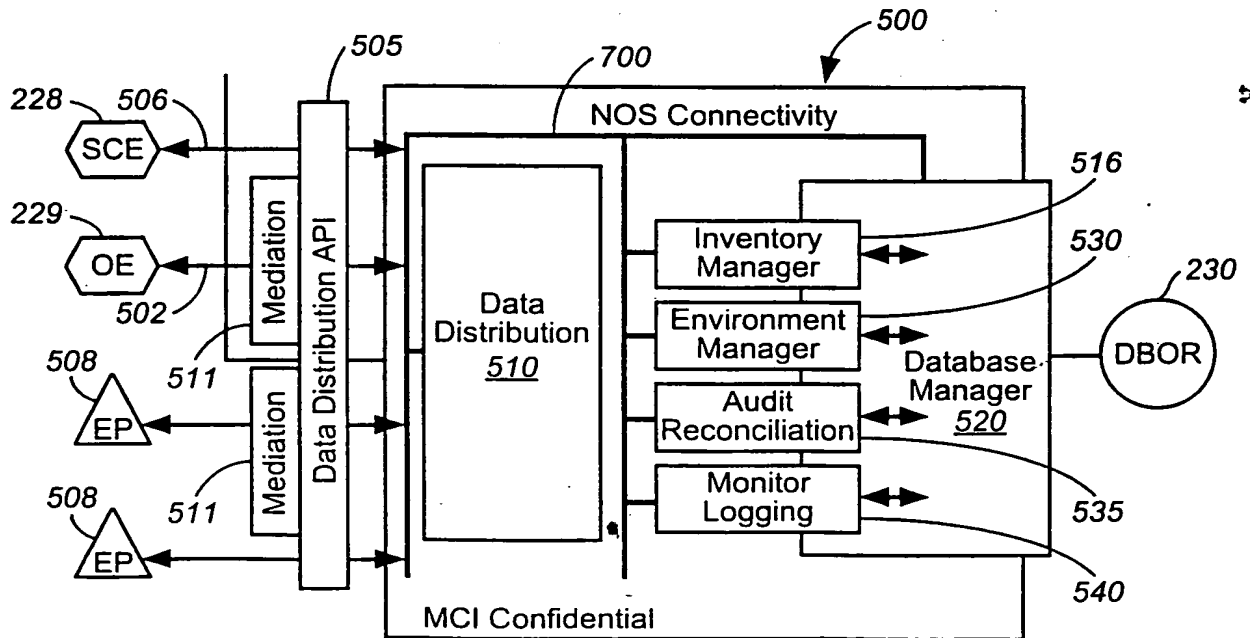


FIG. 5c

6/61

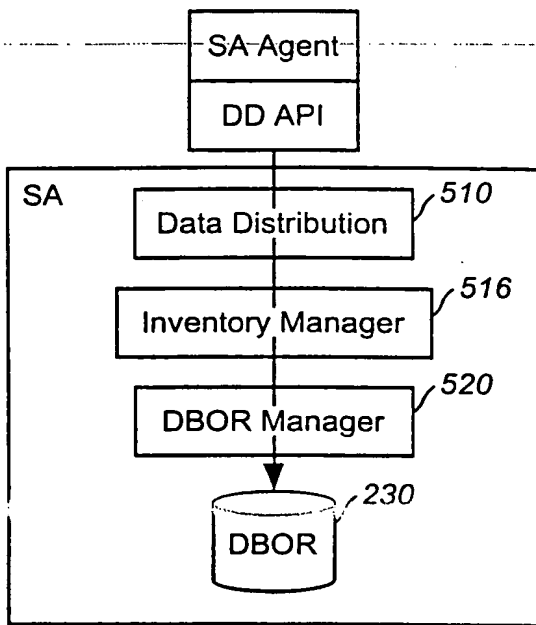


FIG. 5d

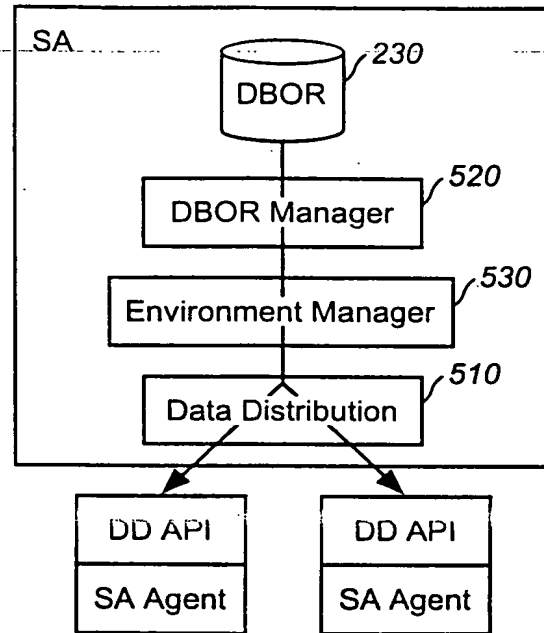


FIG. 5e

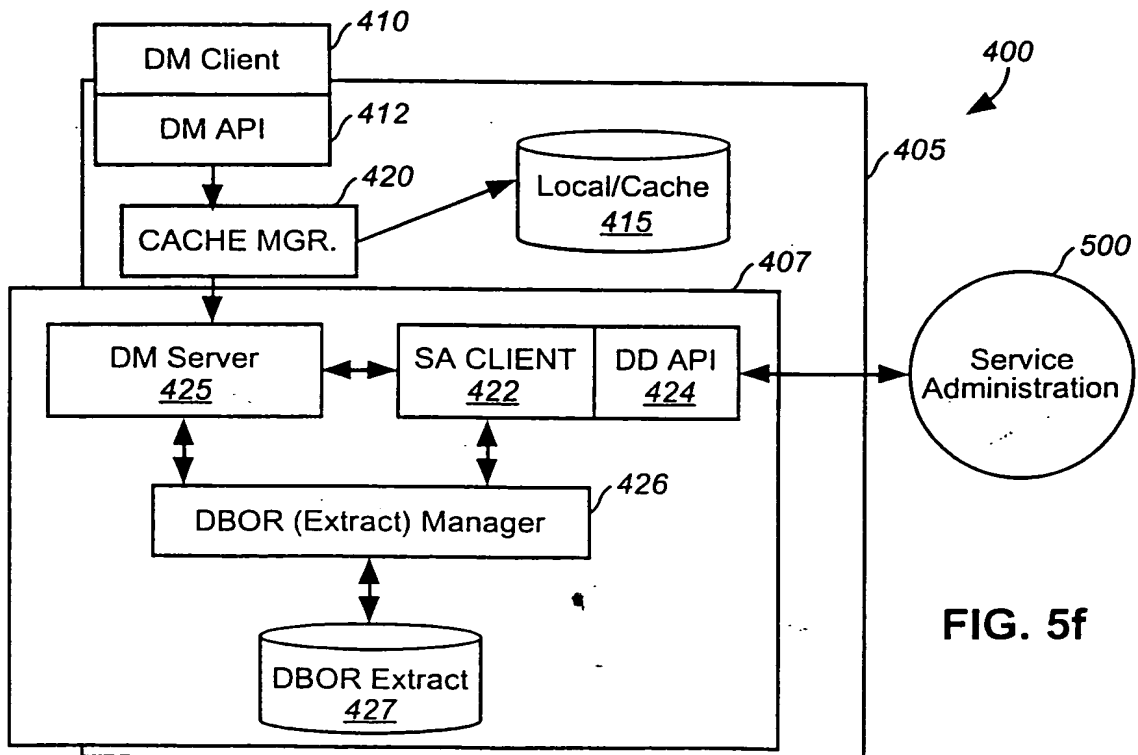
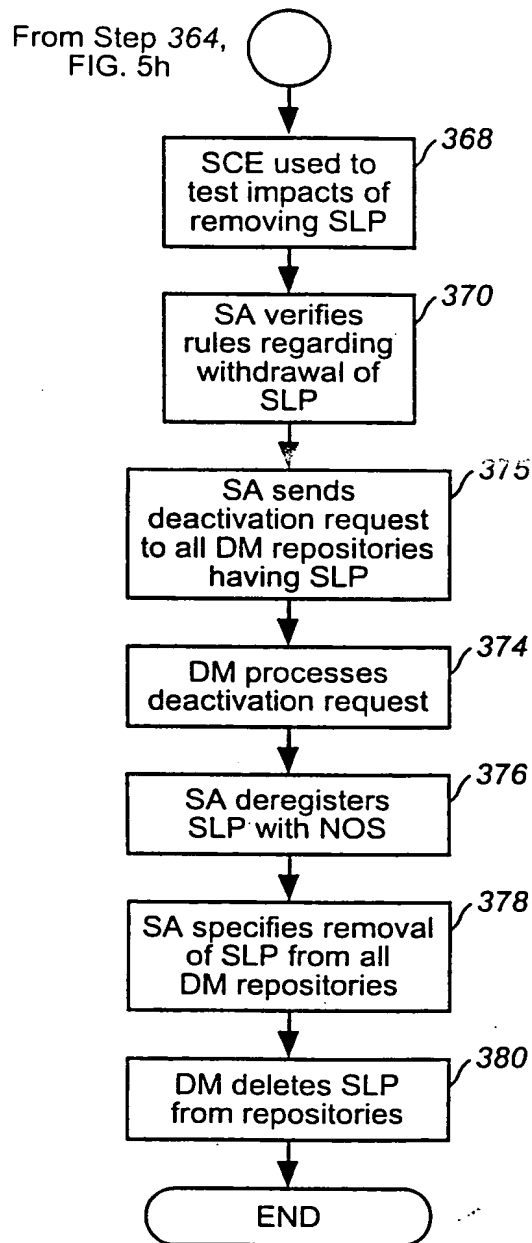
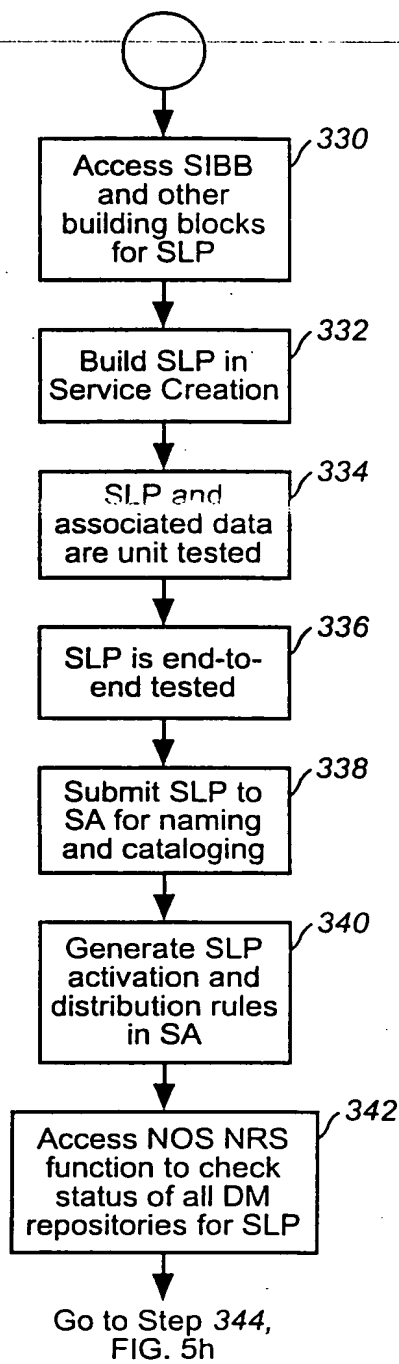


FIG. 5f

7/61



8/61

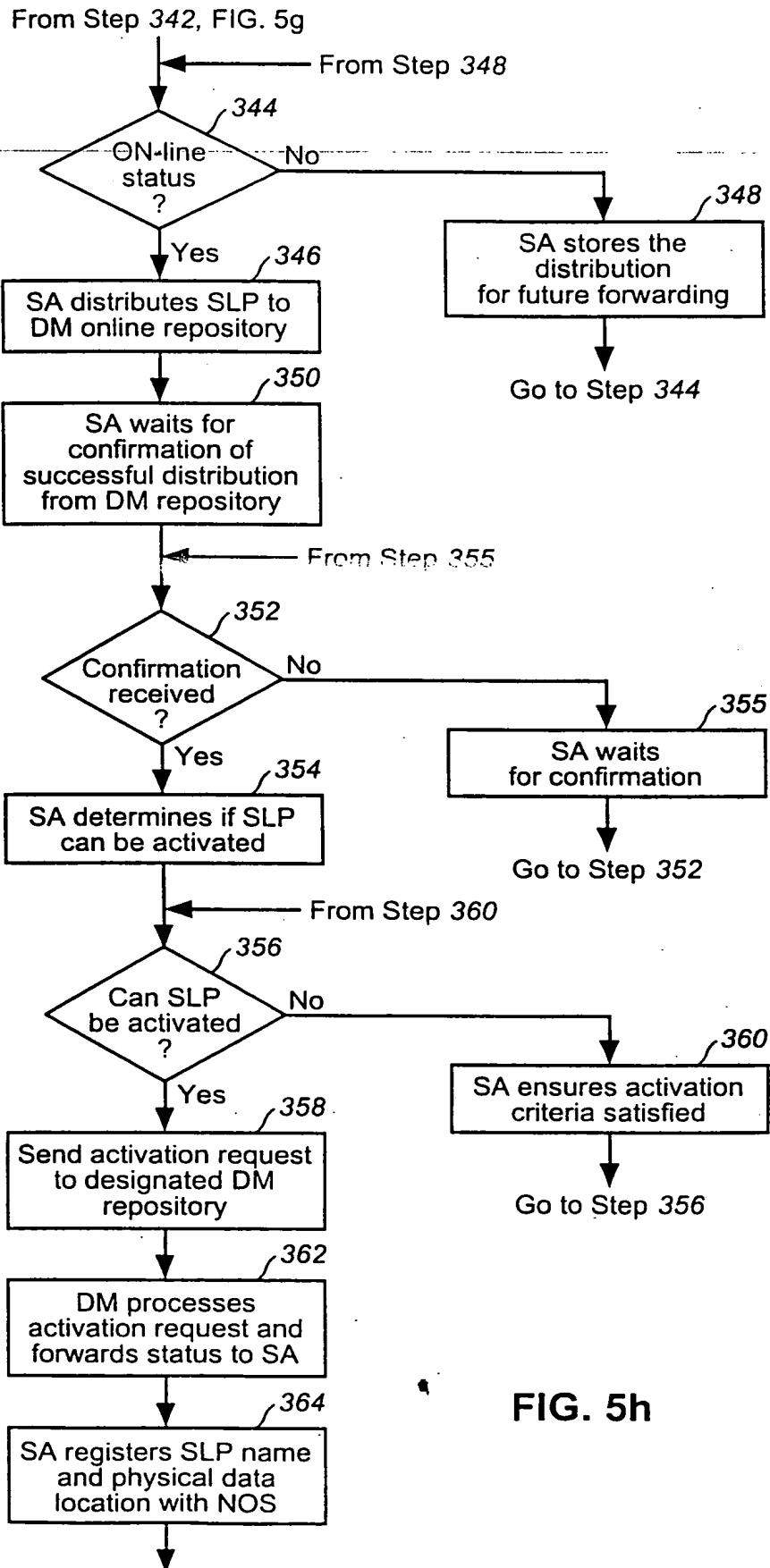


FIG. 5h

10/61

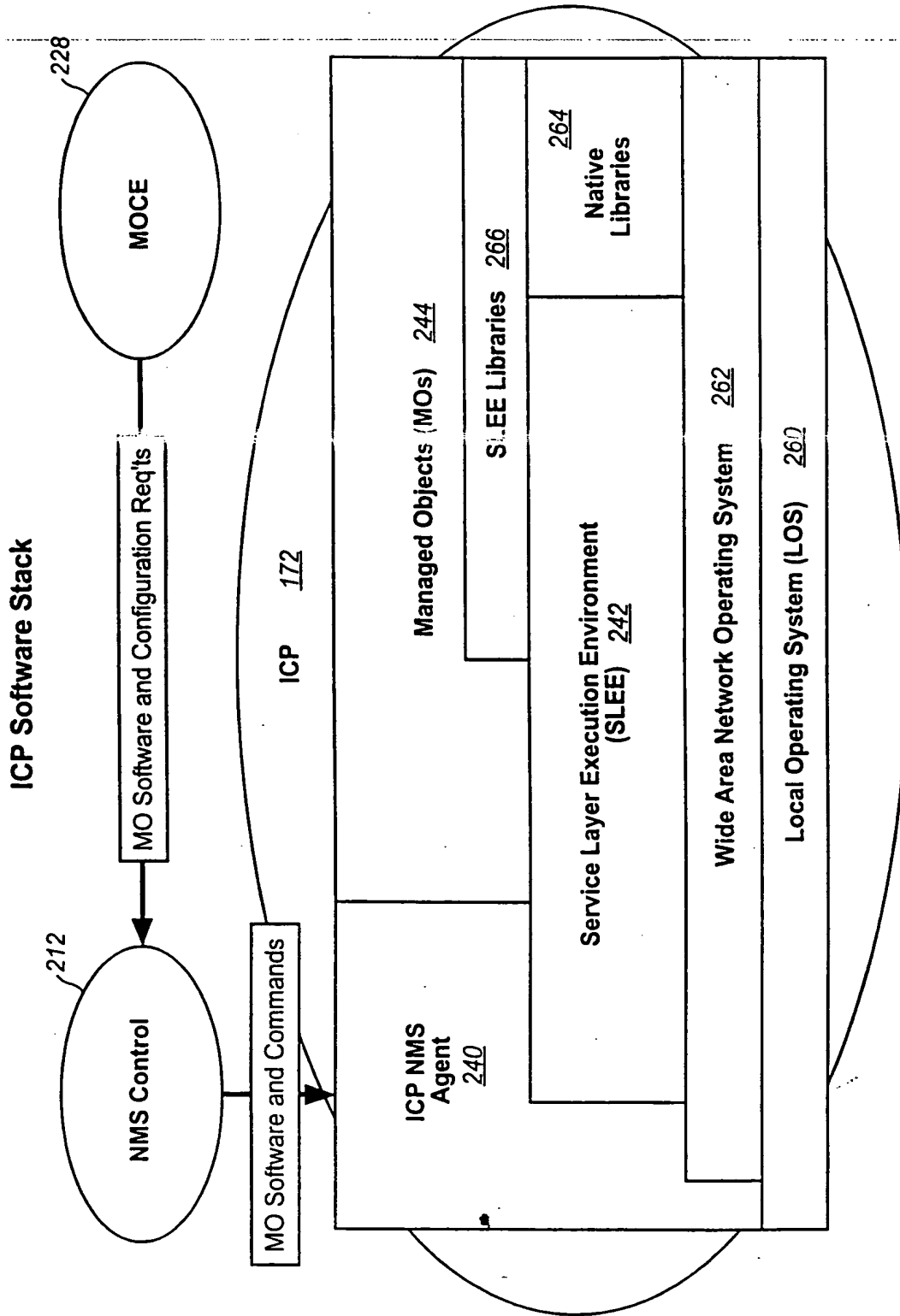


FIG. 7

11/61

VM to SLEE

Local Operating System 270

Instance of VM as a Process 270

SLEE Management Code as Main Program 272

Class Table 276

Managed Object Instances 274

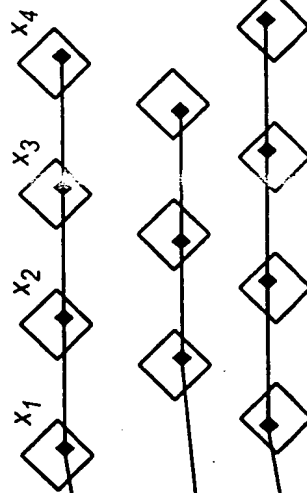


FIG. 8

12/61

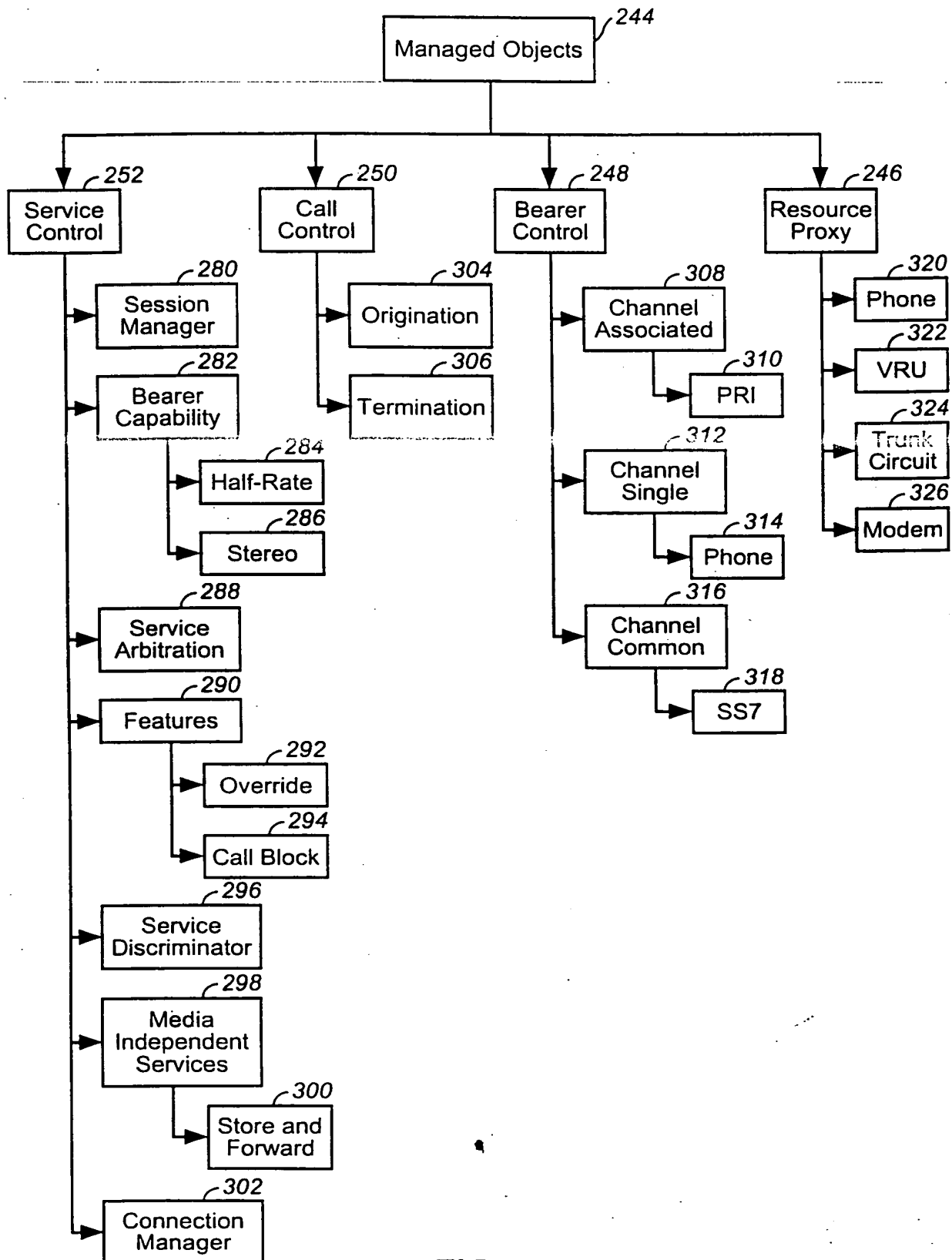


FIG. 9

13/61

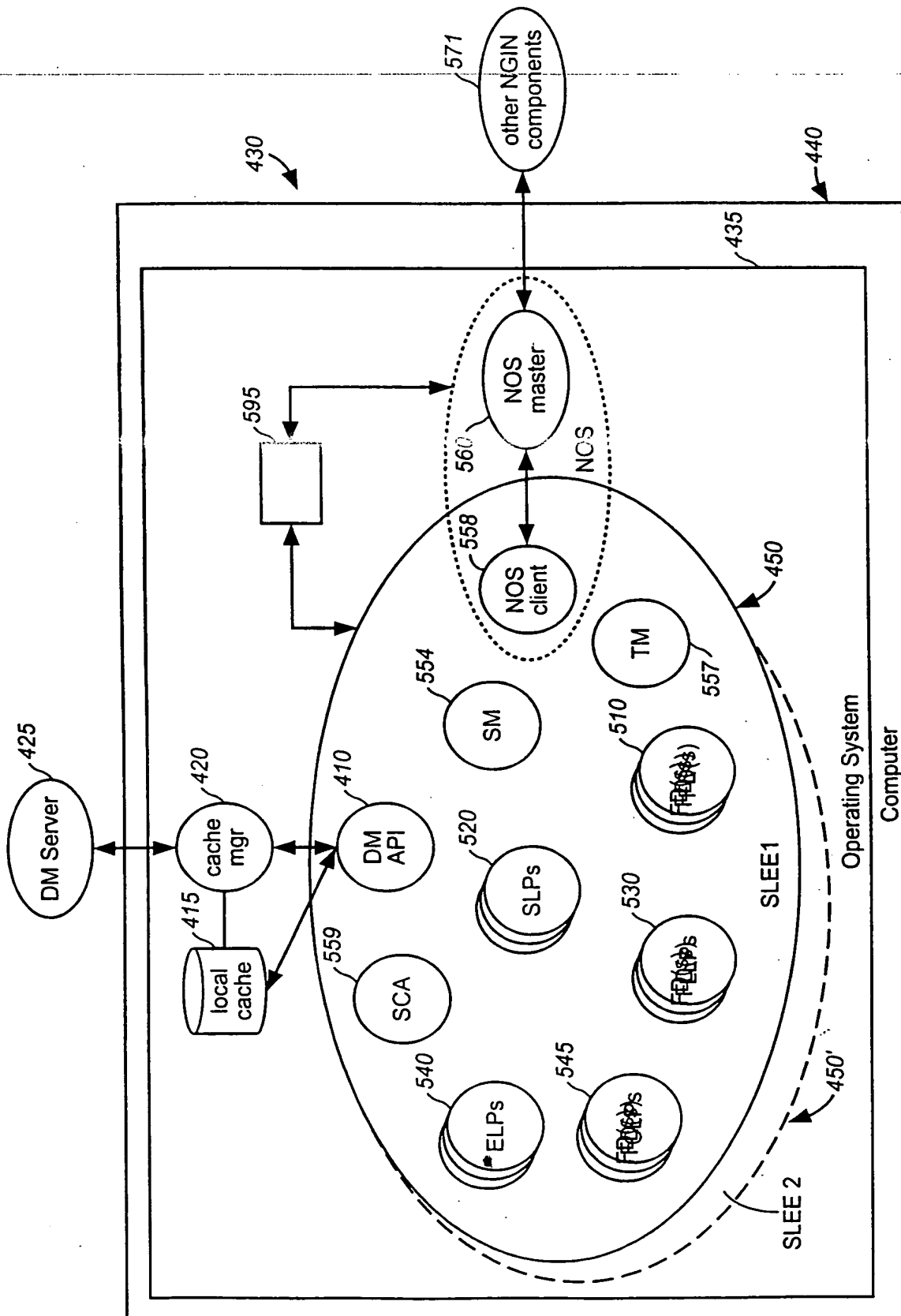


FIG. 10a

14/61

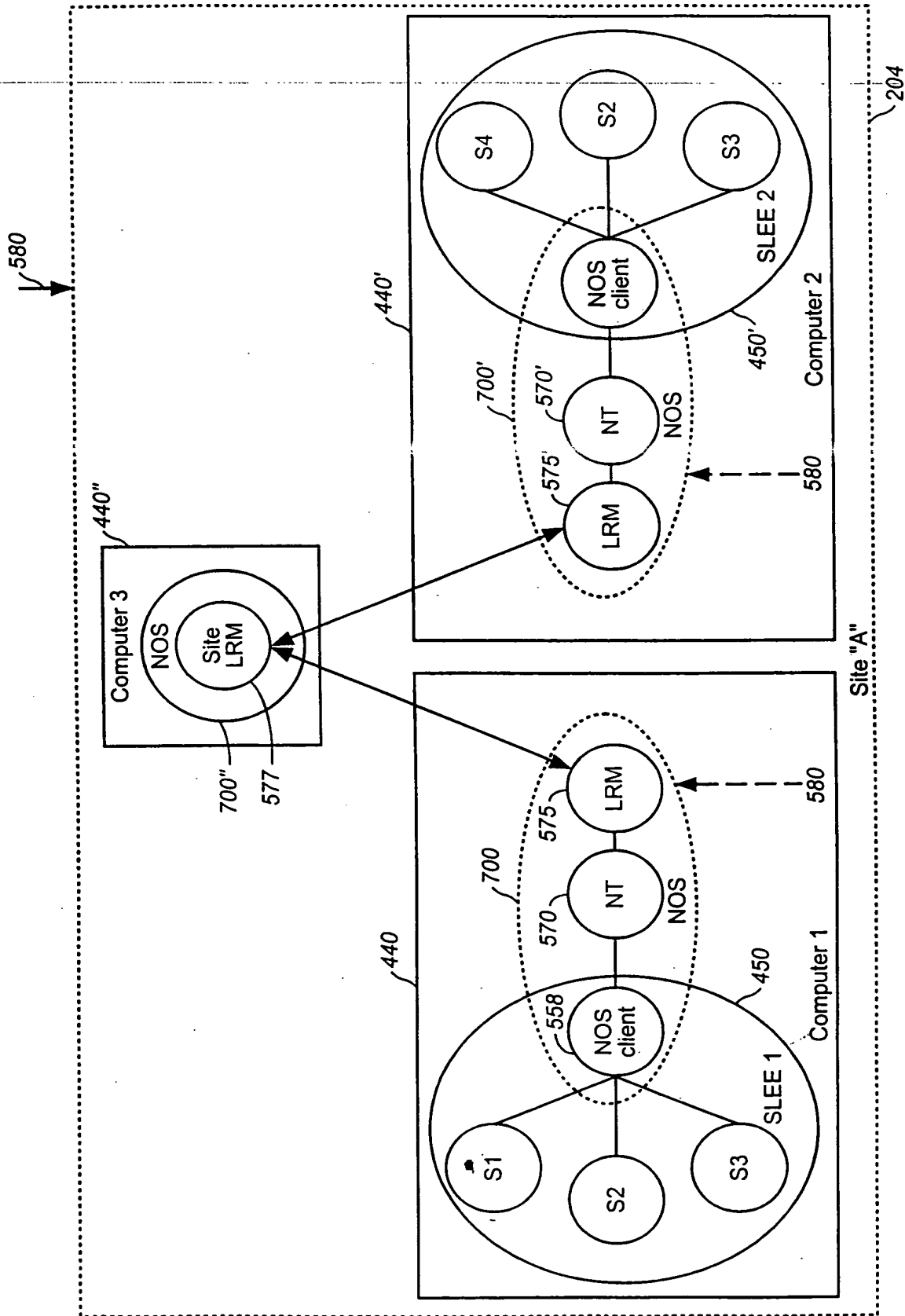


FIG. 10b

15/61

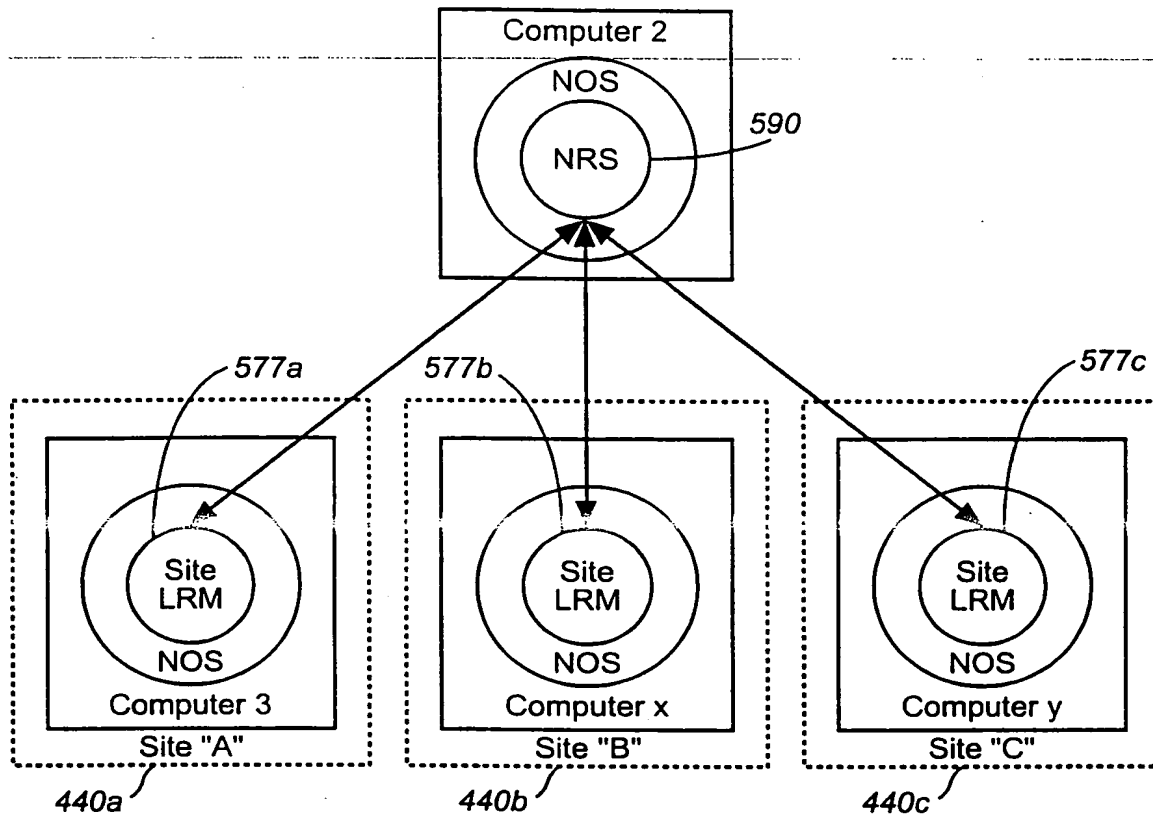


FIG. 10c

16/61

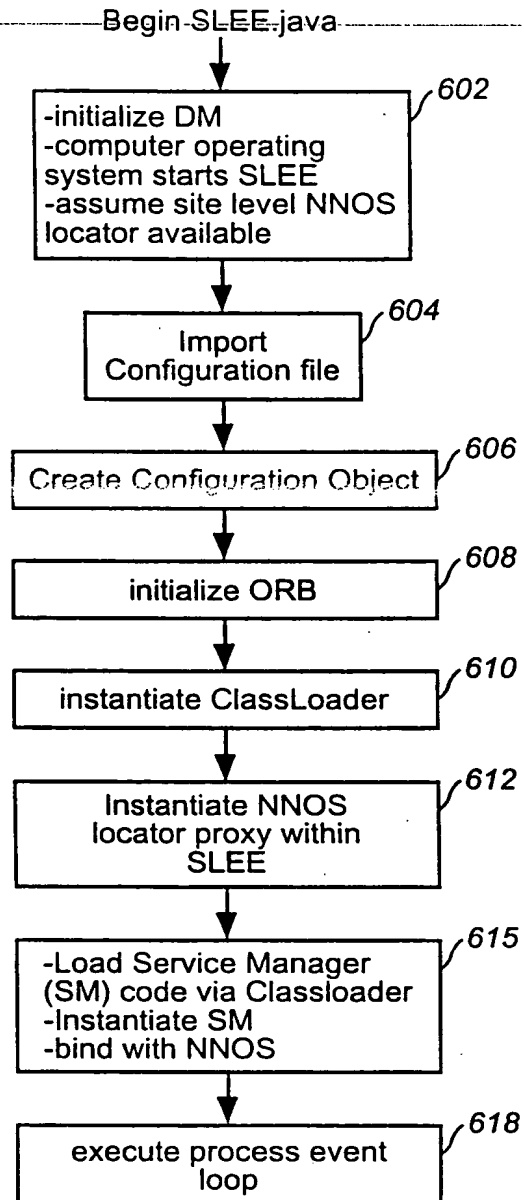


FIG. 11a

17/61

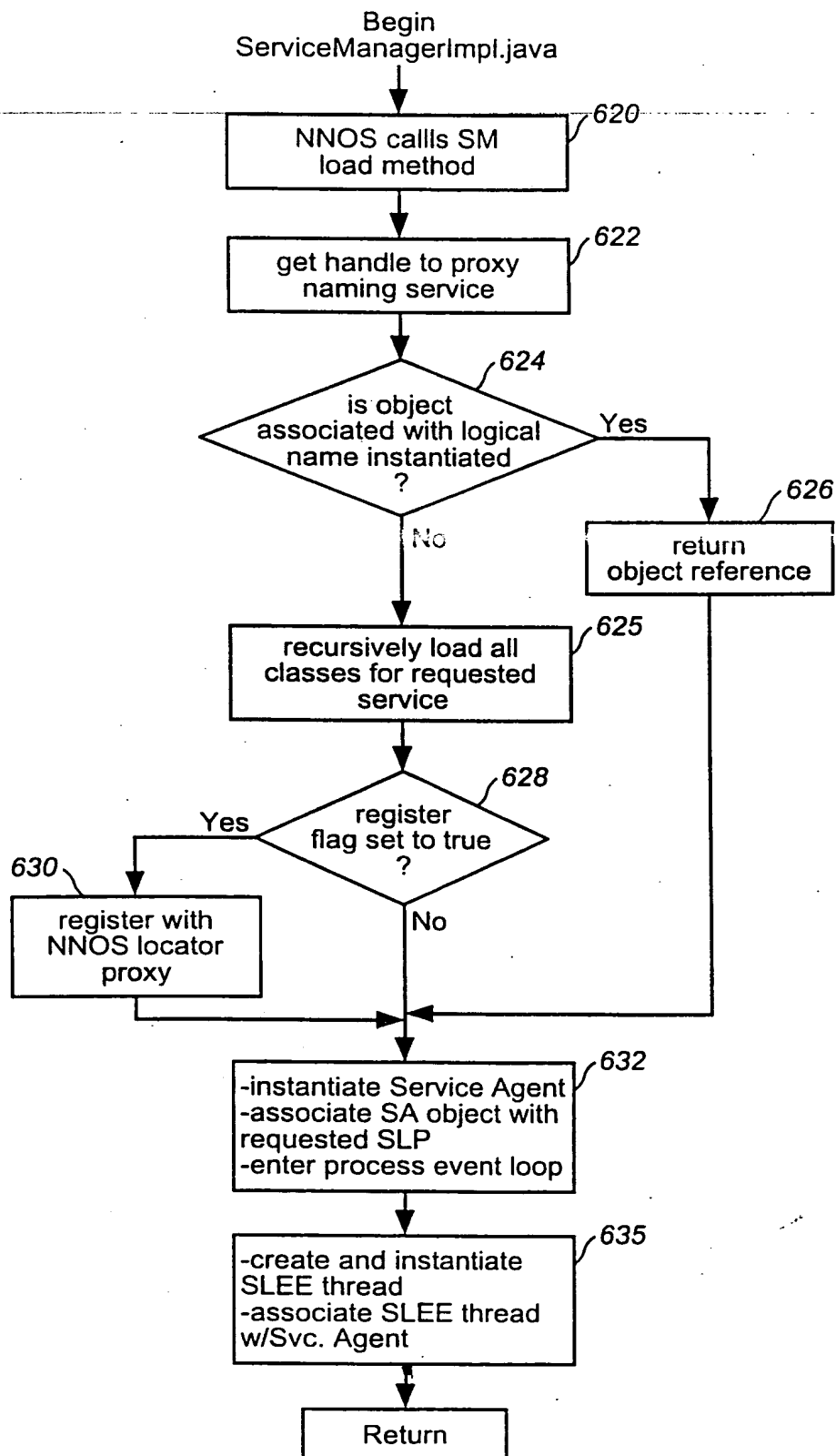


FIG. 11b

18/61

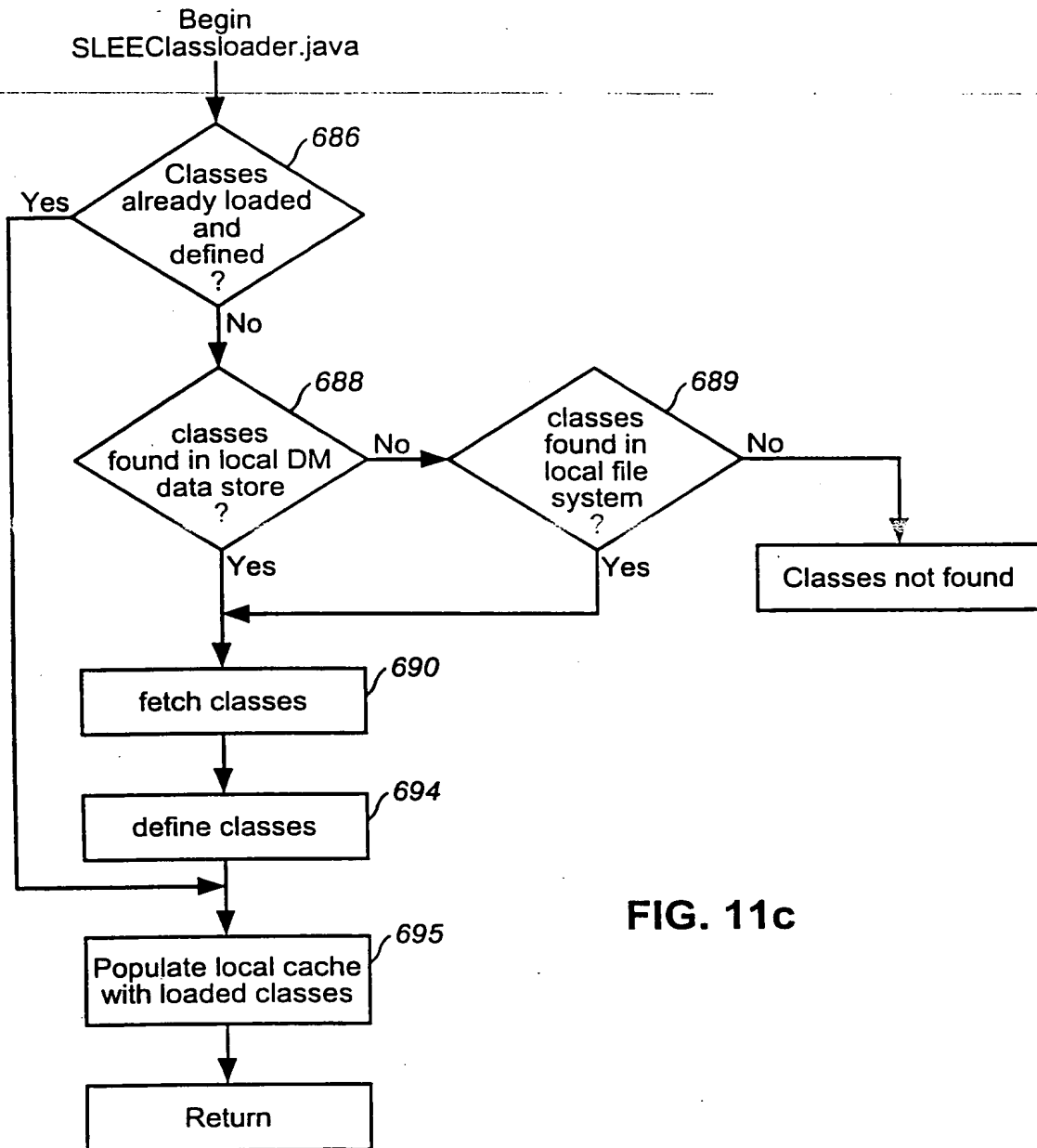
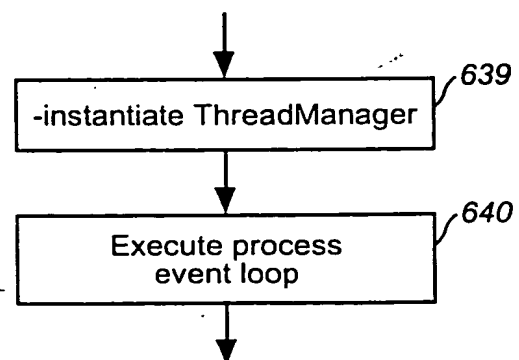


FIG. 11c

FIG. 11d



19/61

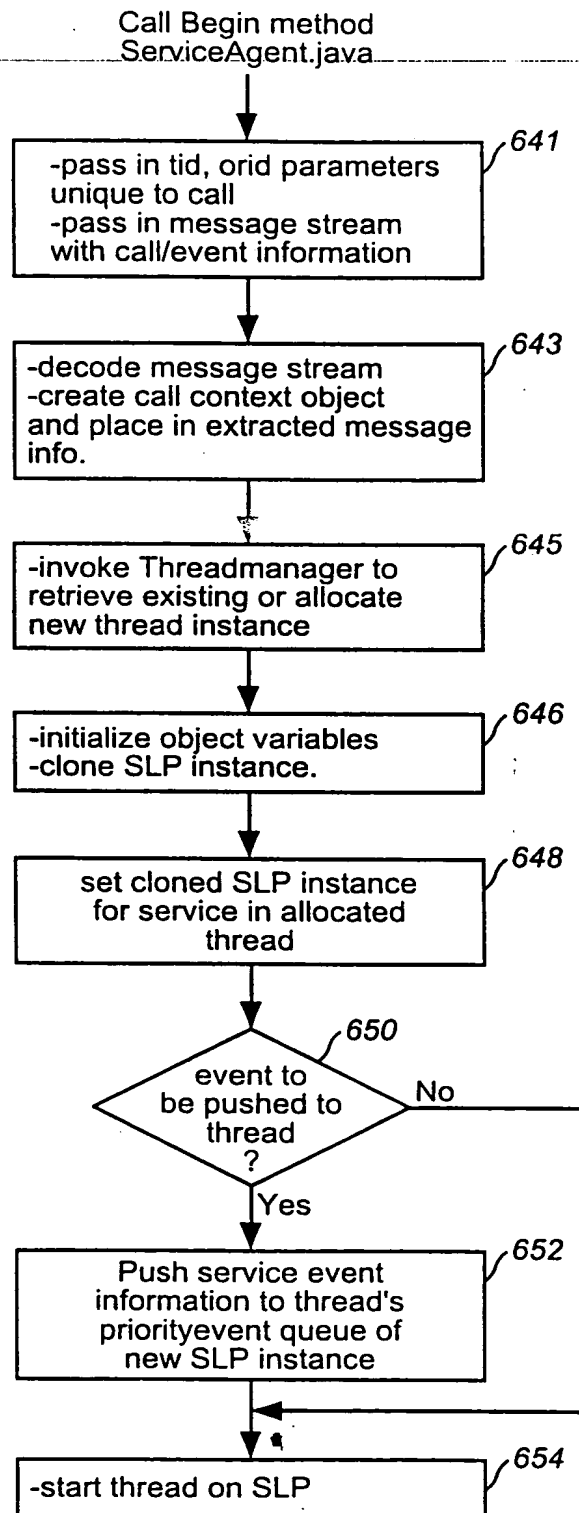


FIG. 11e

20/61

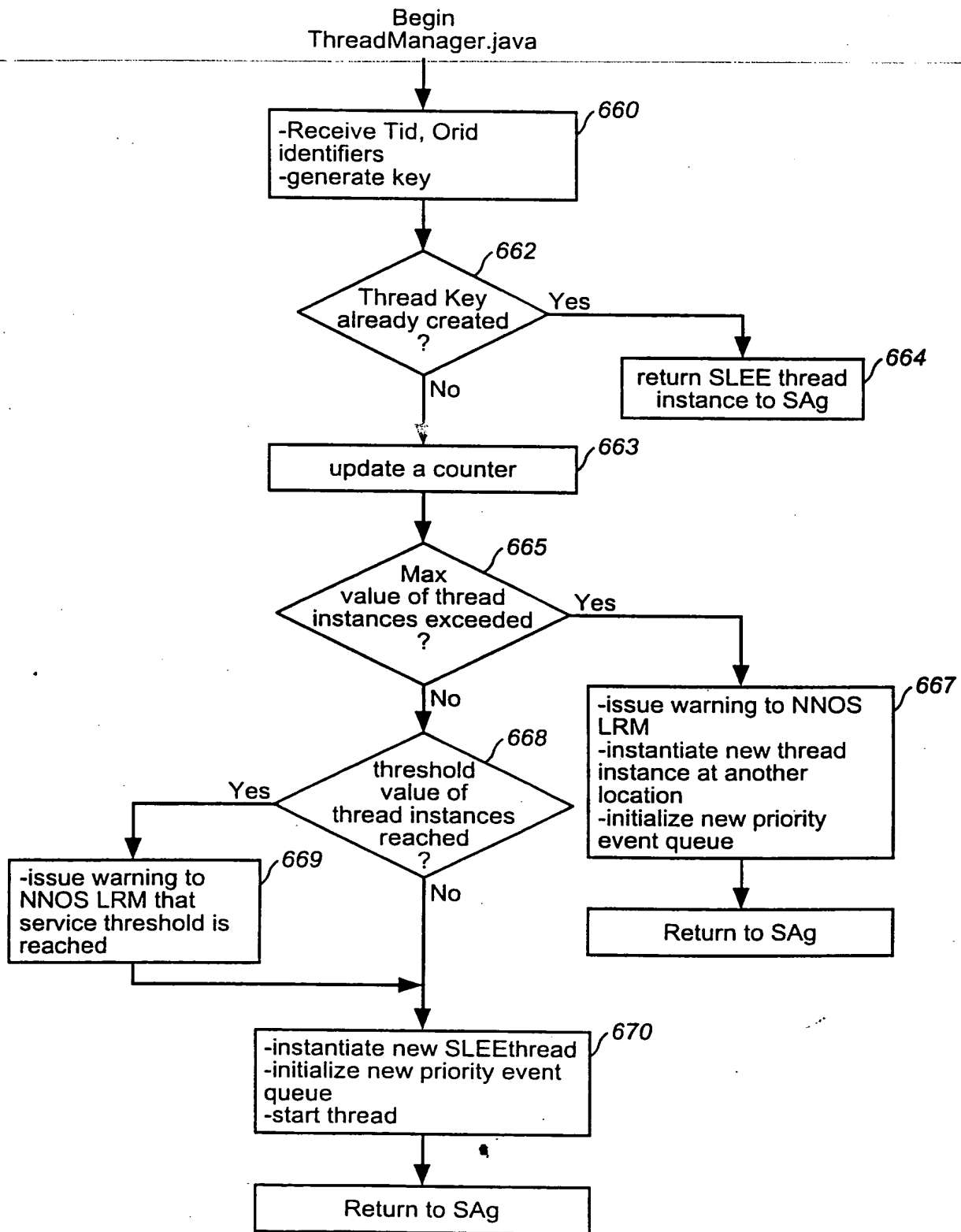


FIG. 11f

21/61

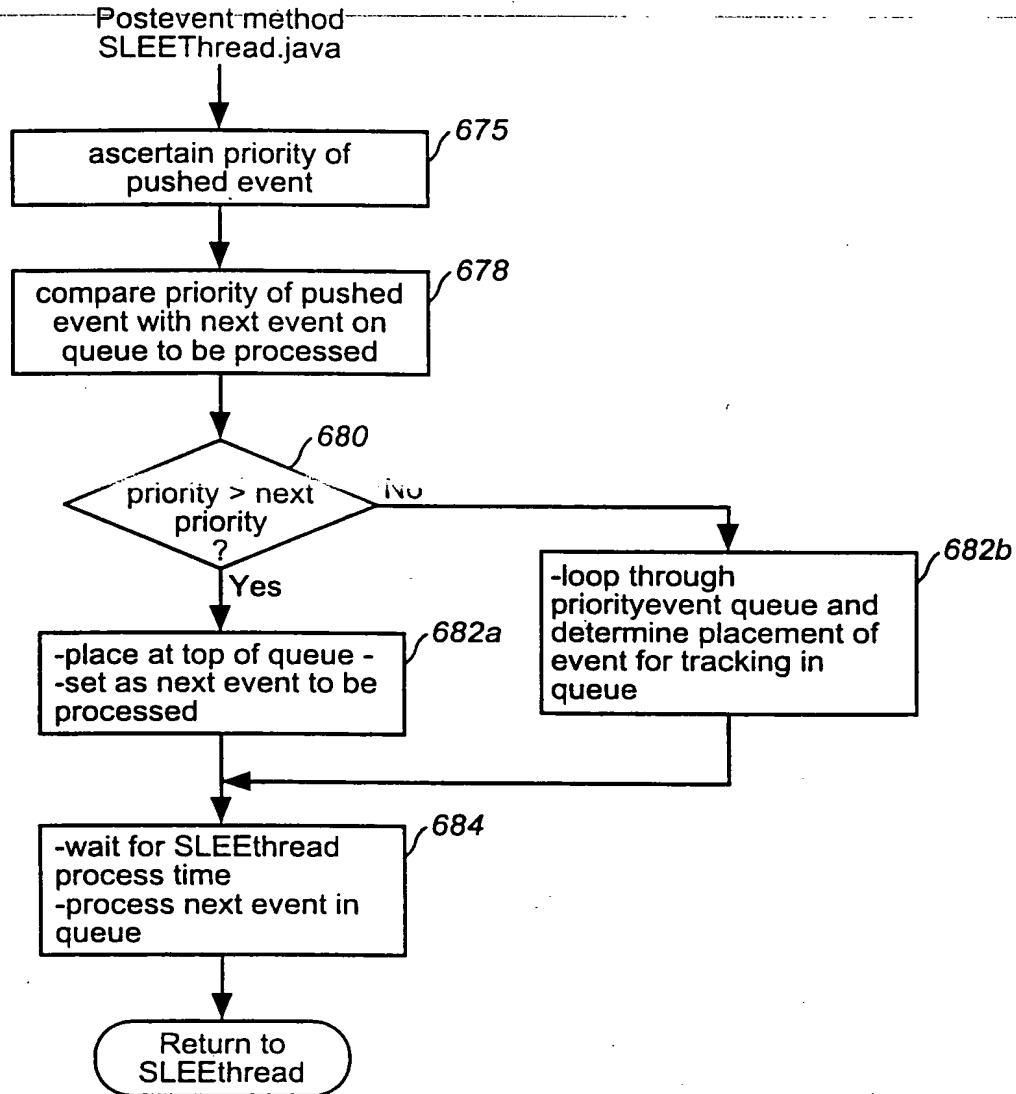


FIG. 11g

22/61

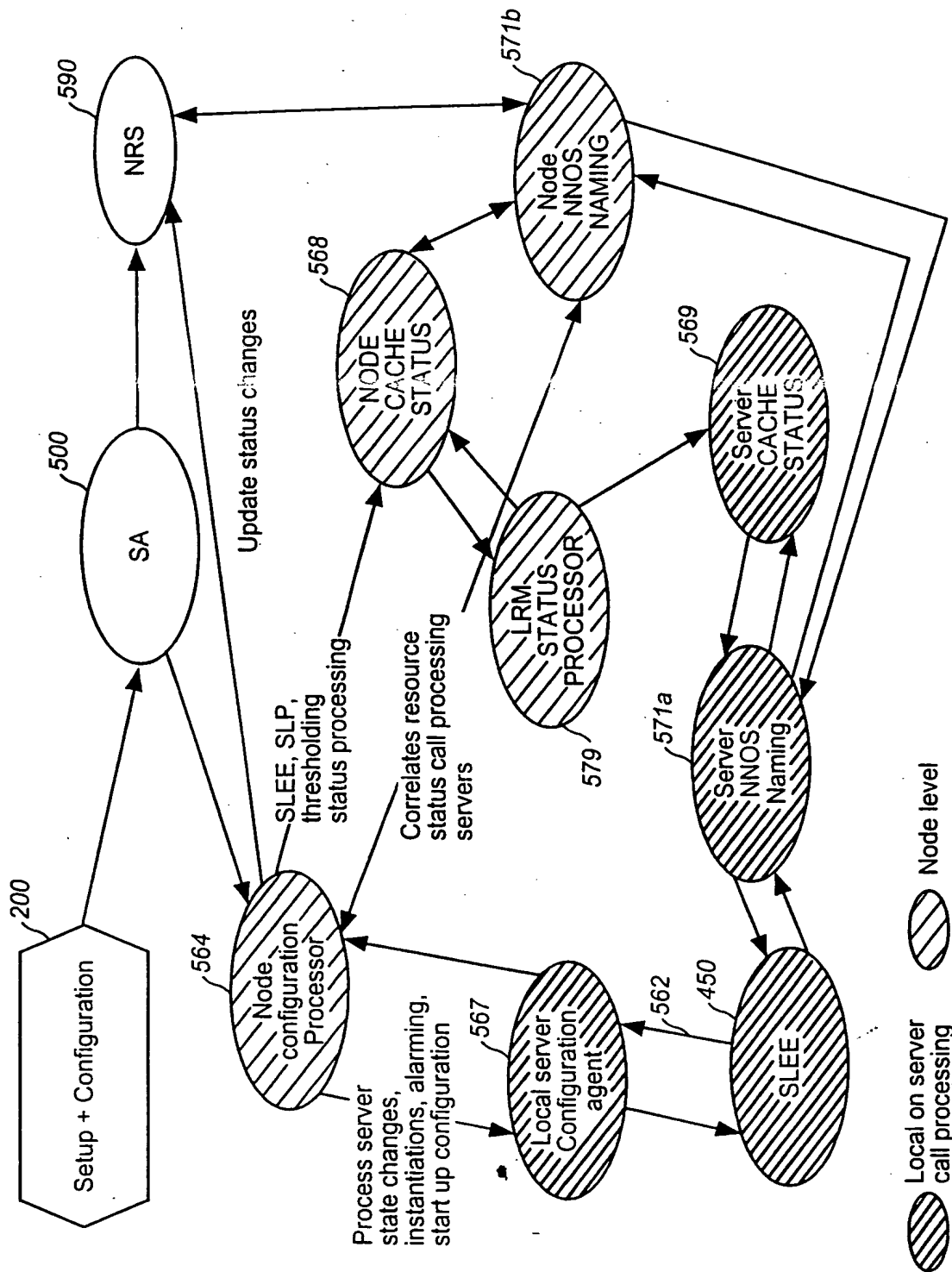
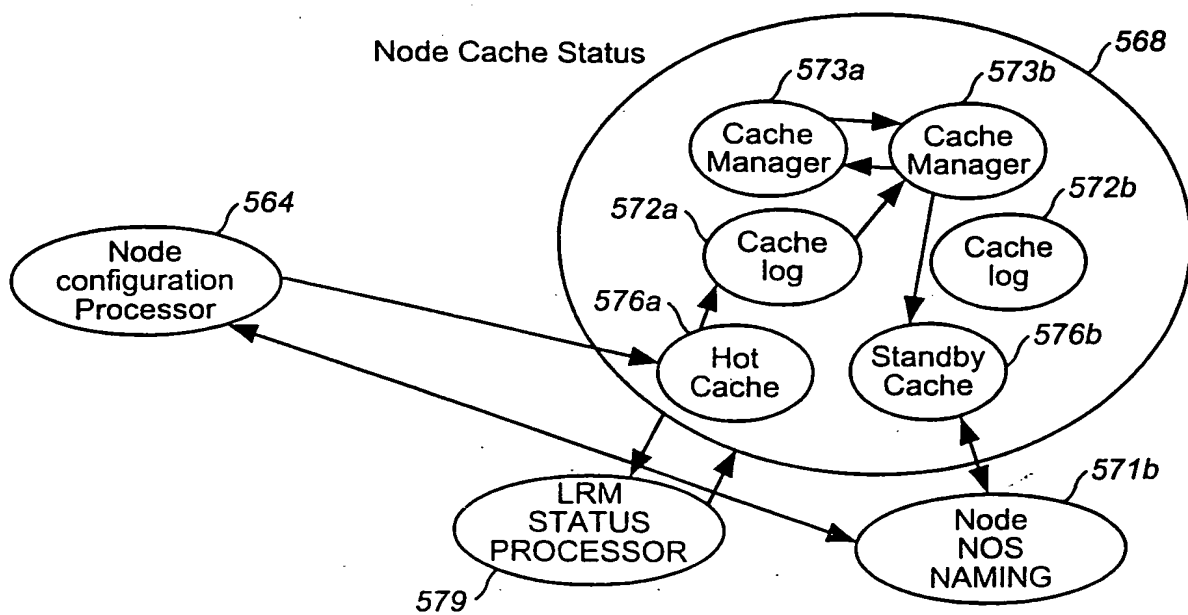
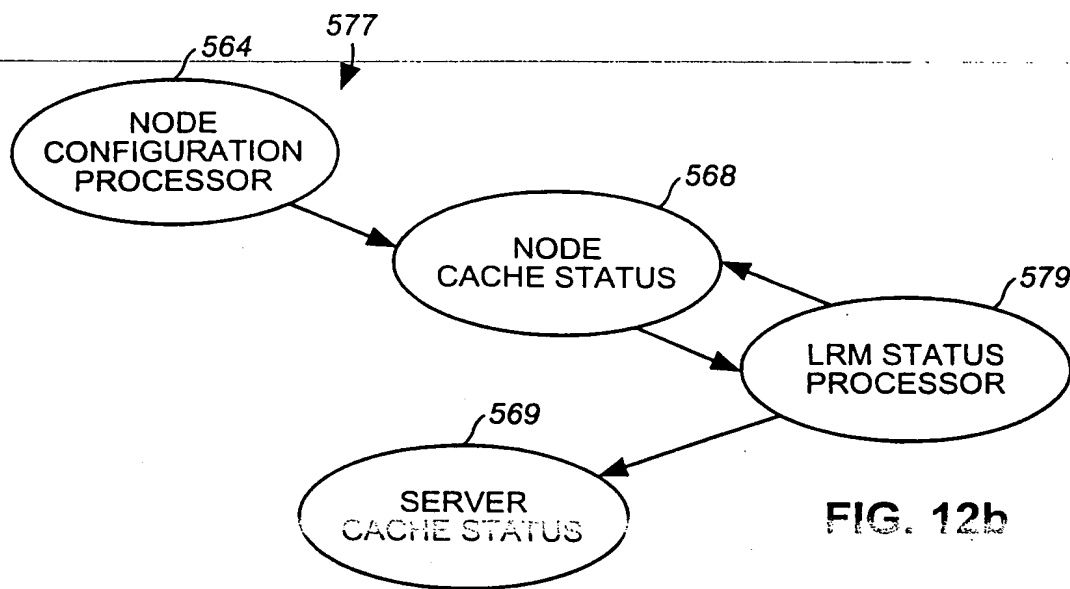
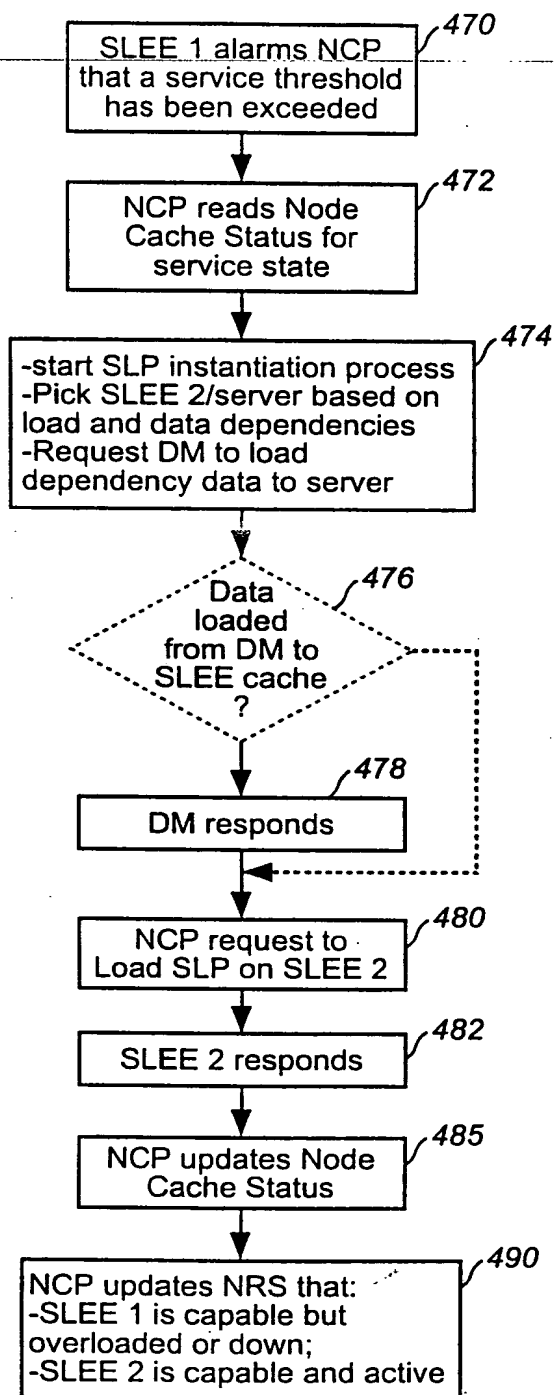
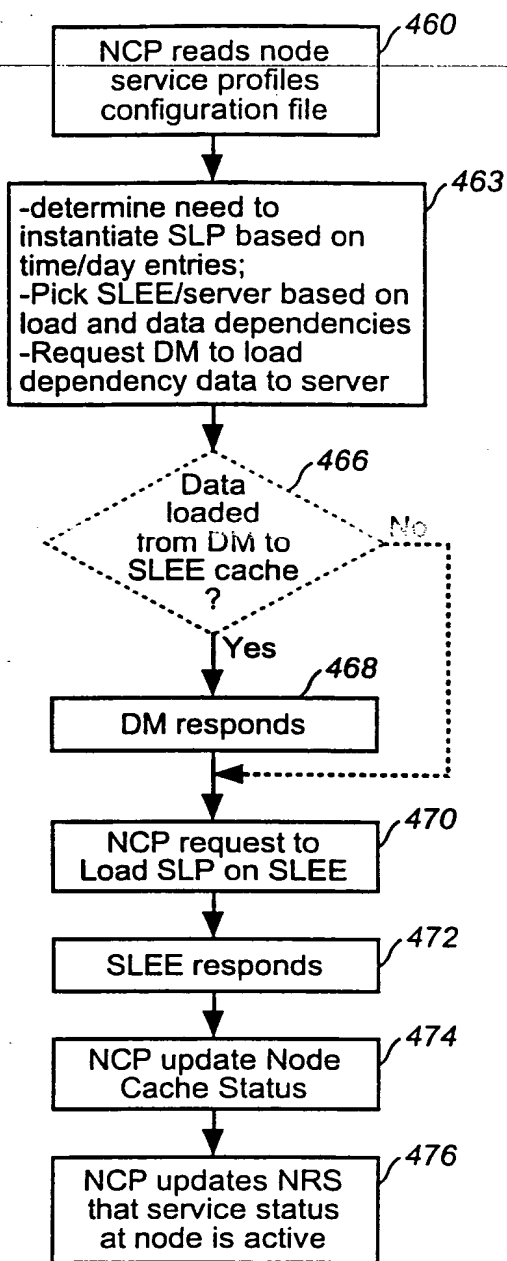


FIG. 12a

23/61



24/61



25/61

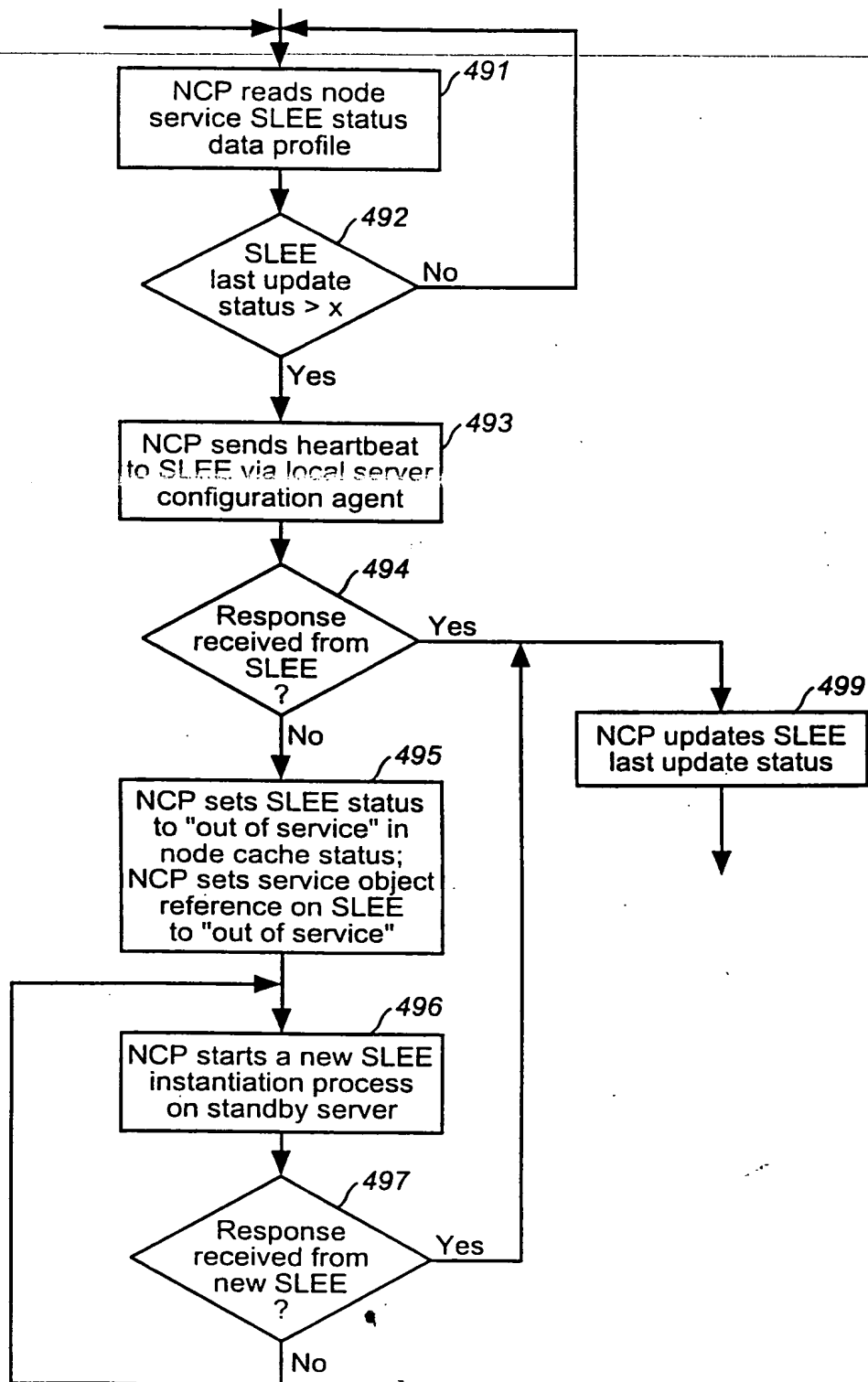
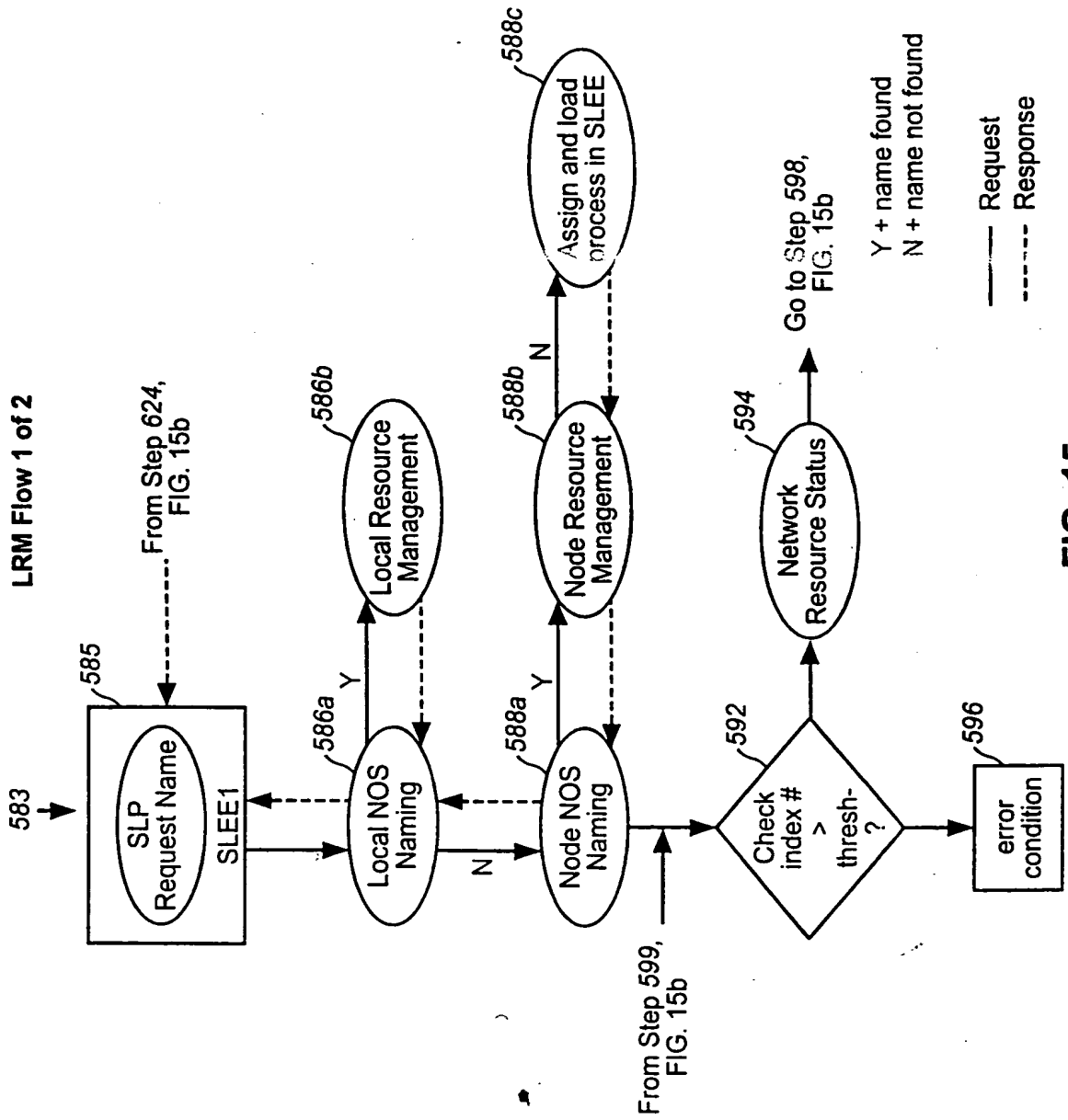


FIG. 14b

26/61



27/61

LRM Flow 2 of 2

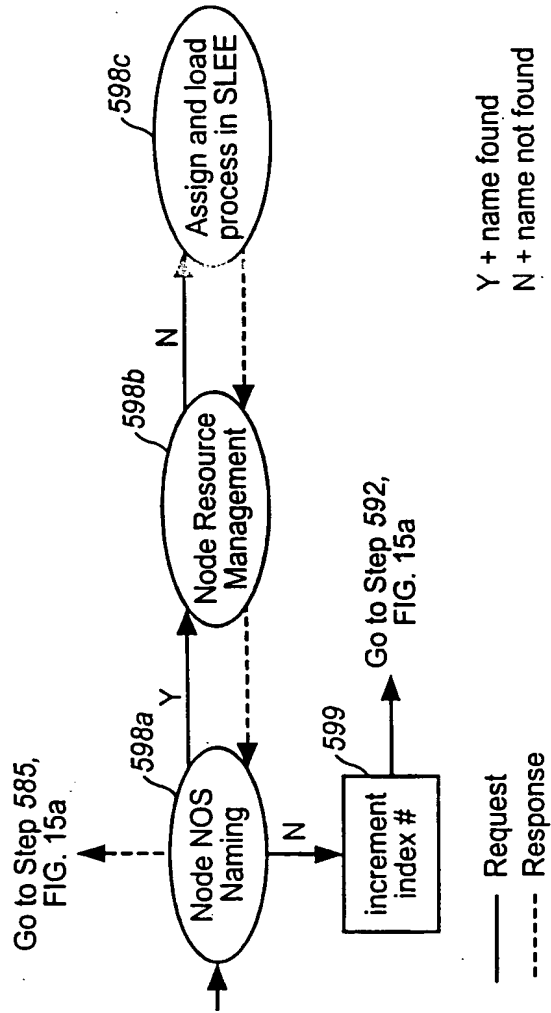


FIG. 15b

28/61

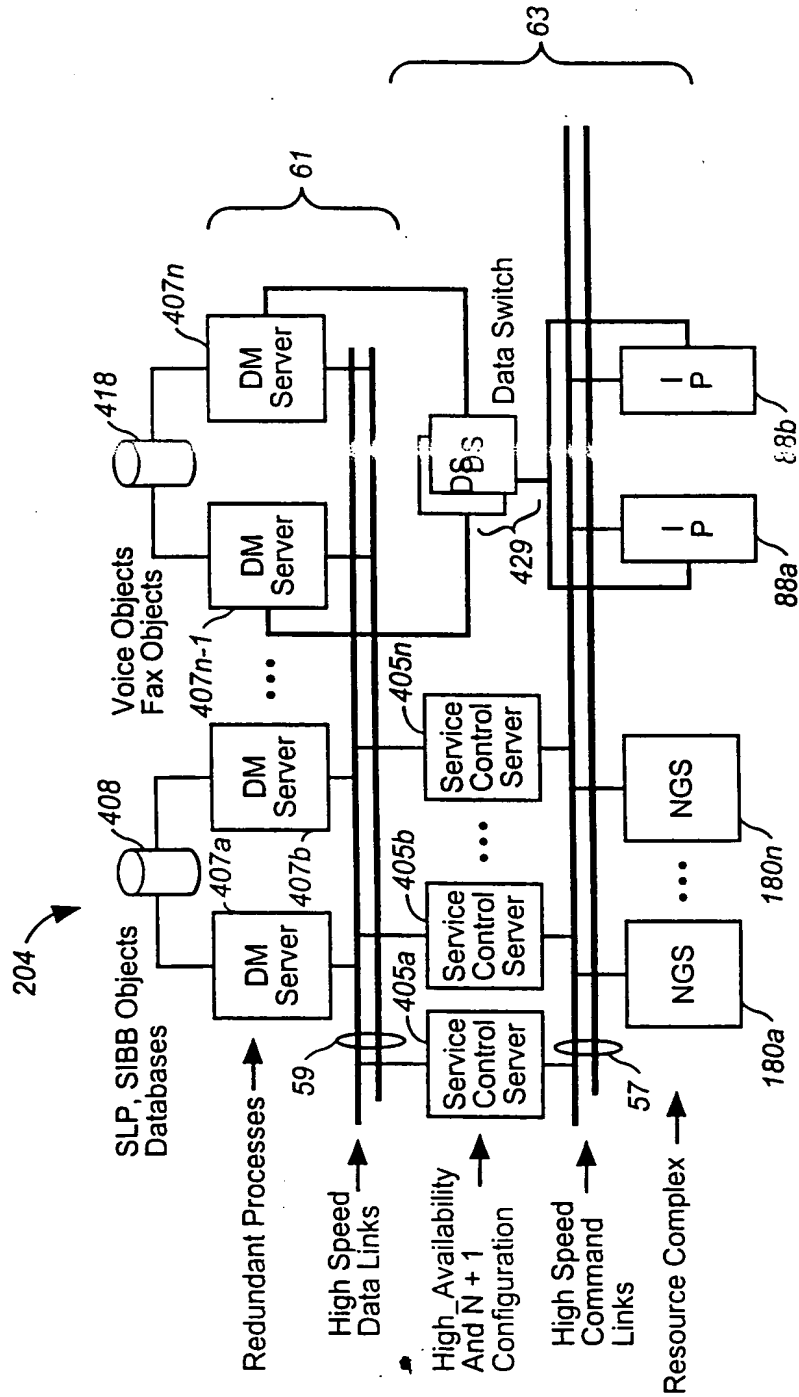


FIG. 16

29/61

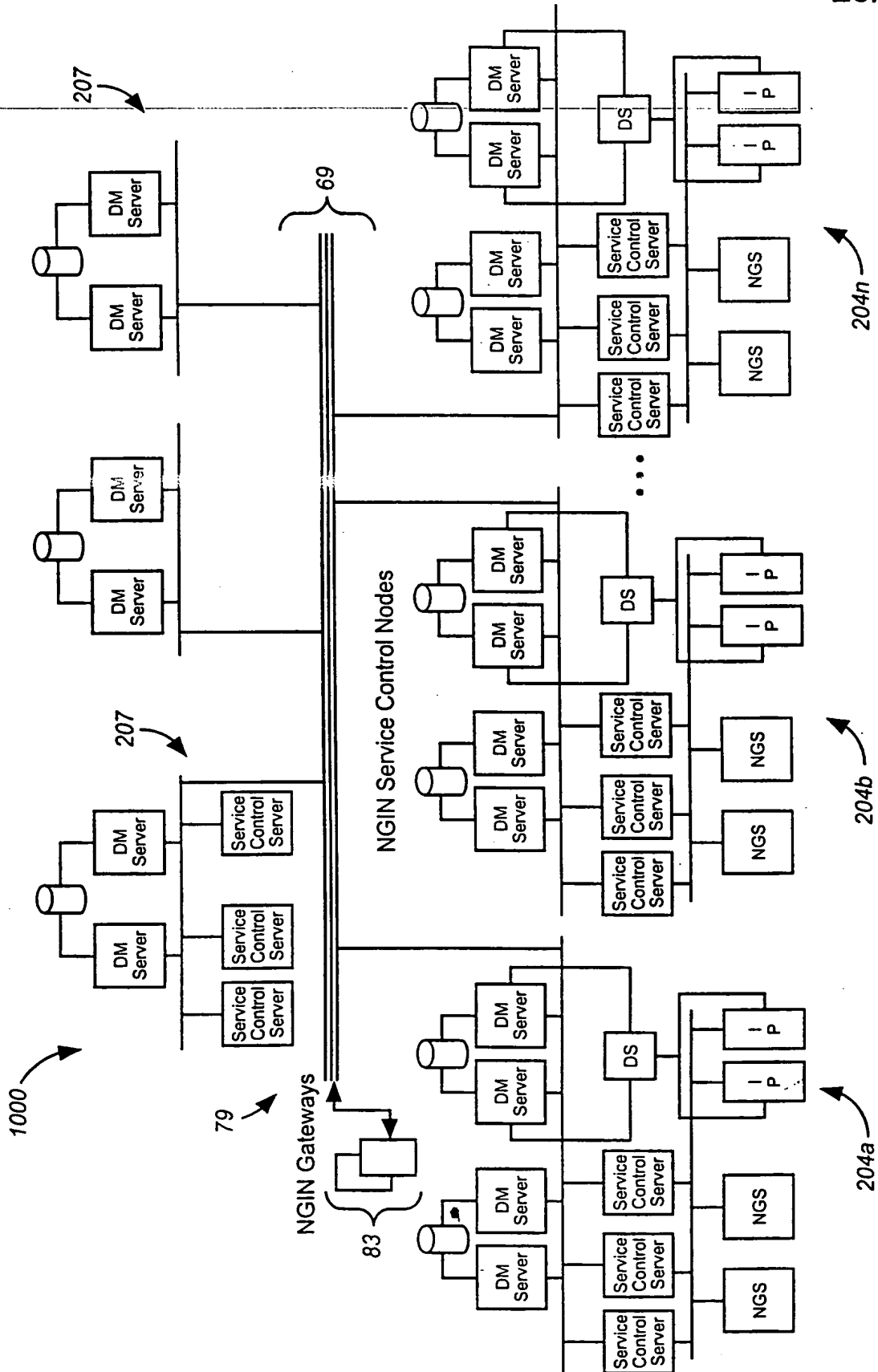
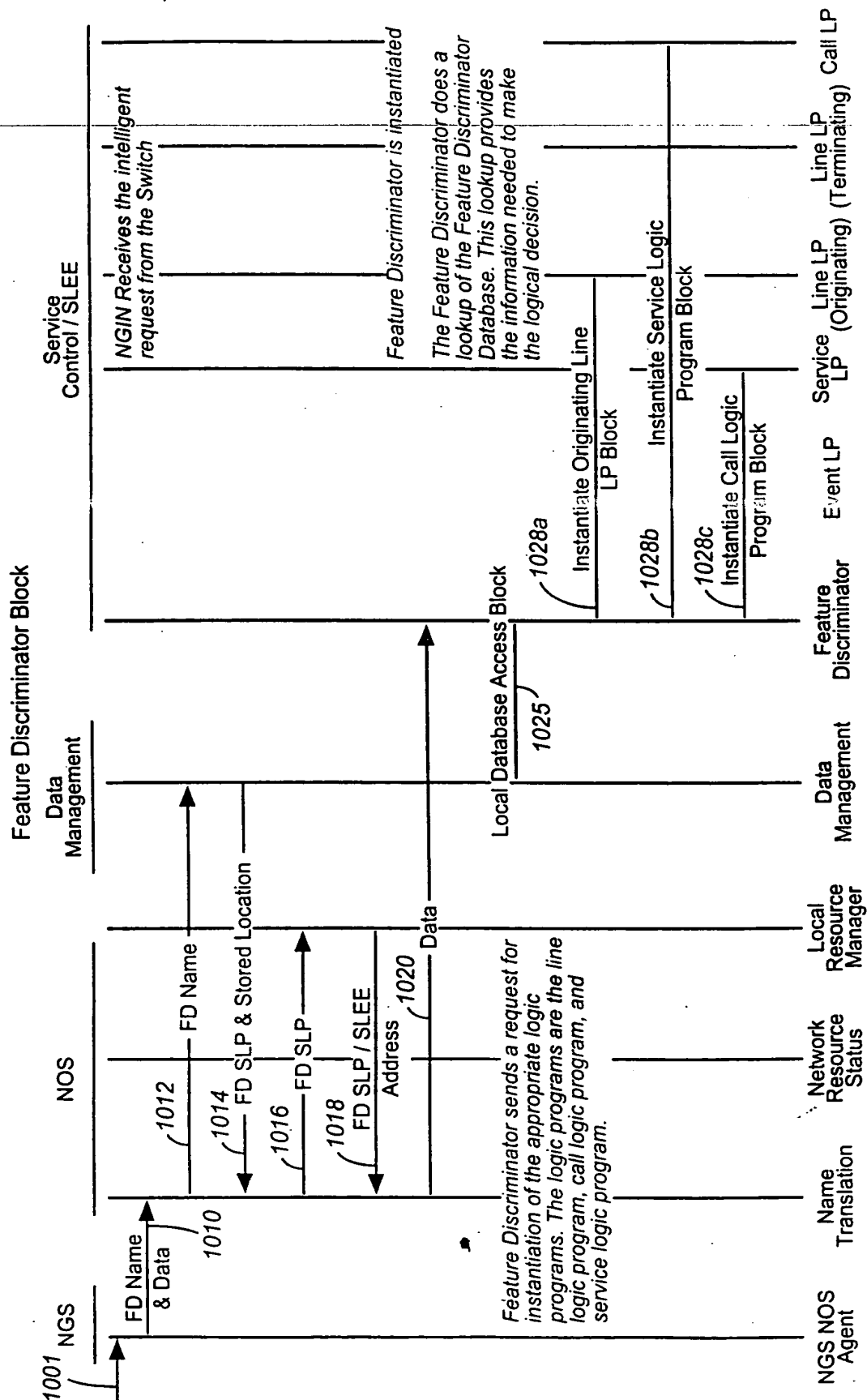


FIG. 17



31/61

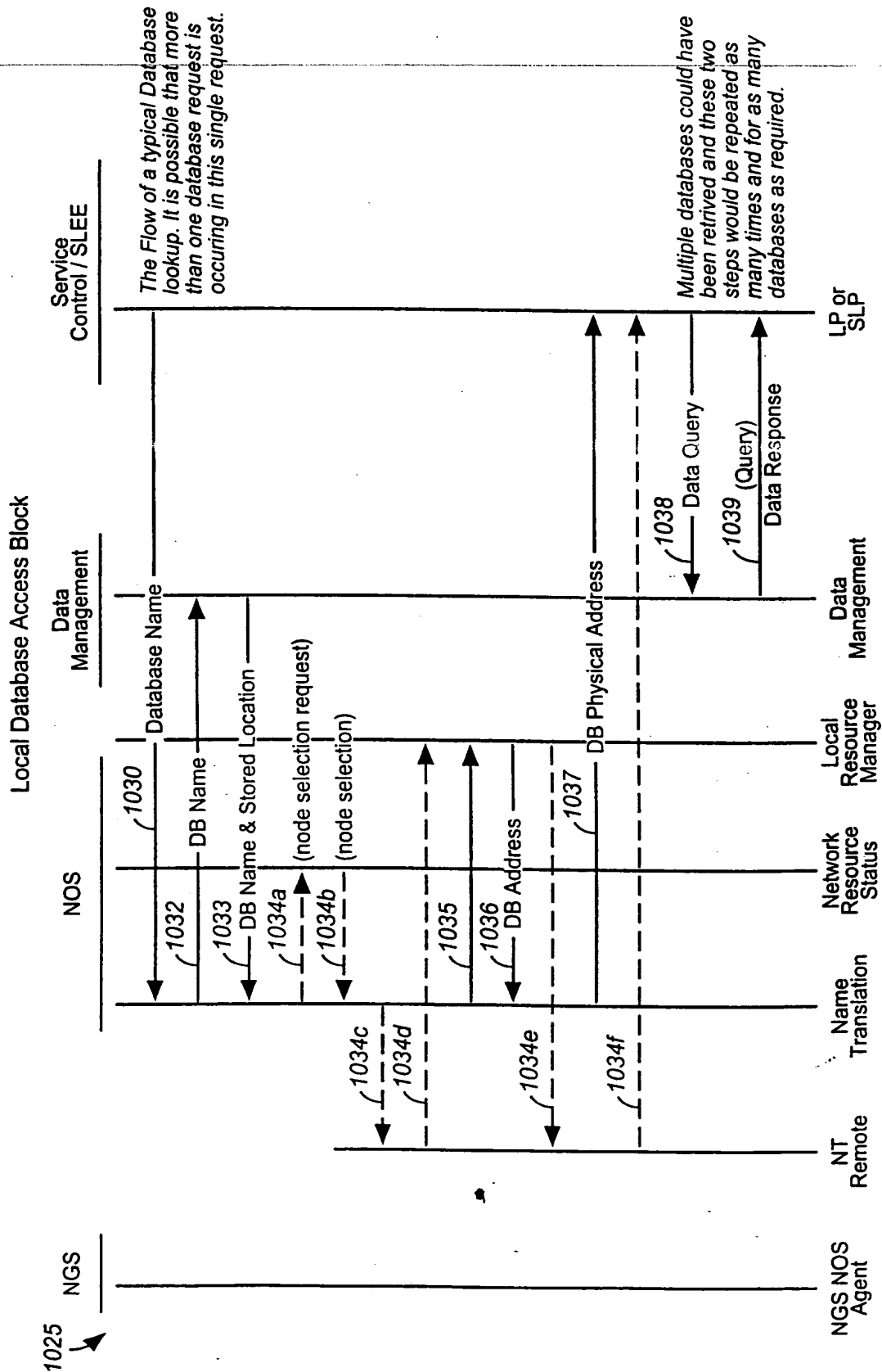
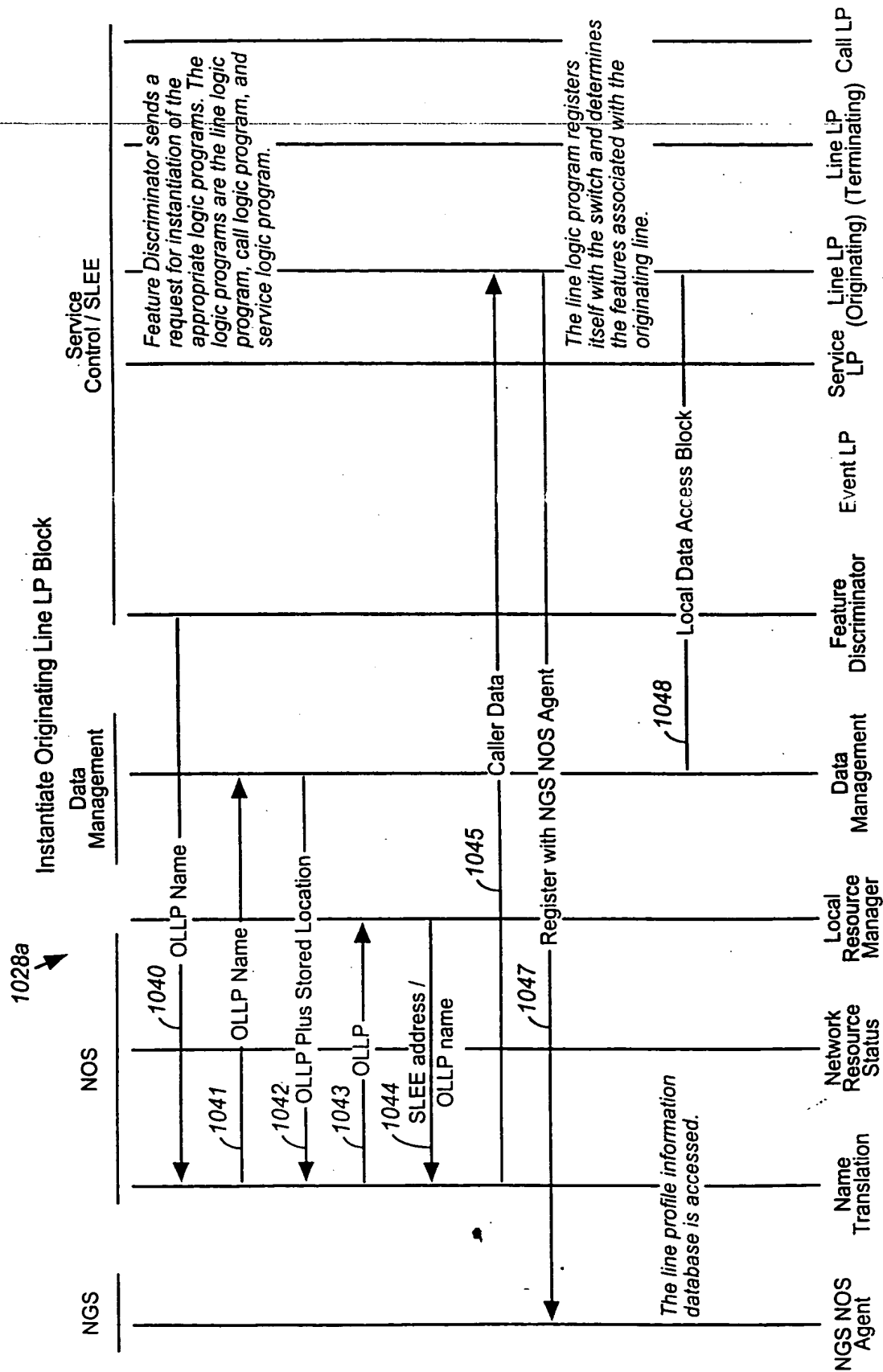


FIG. 18b



32/61

FIG. 18c

33/61

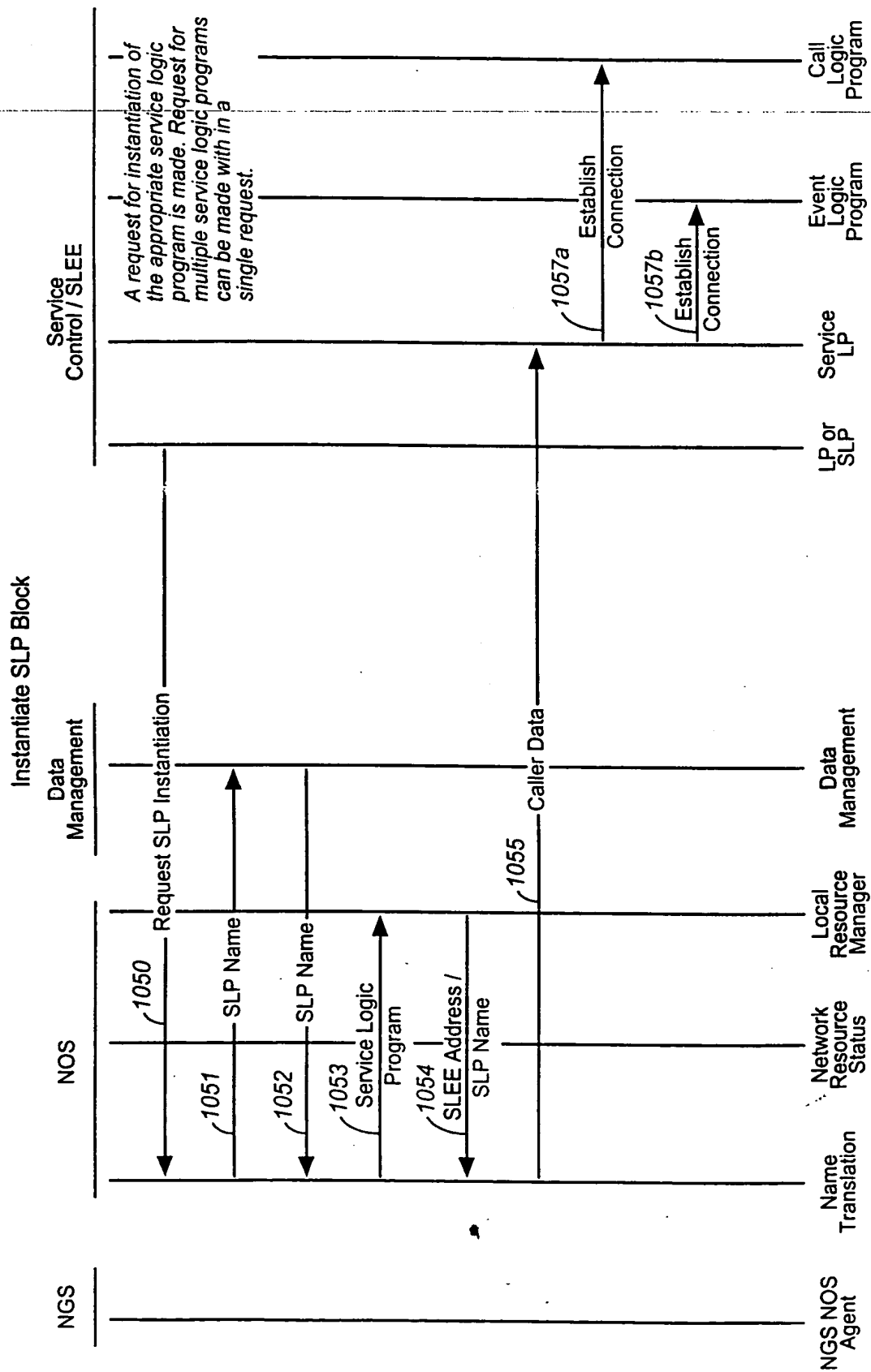


FIG. 18d

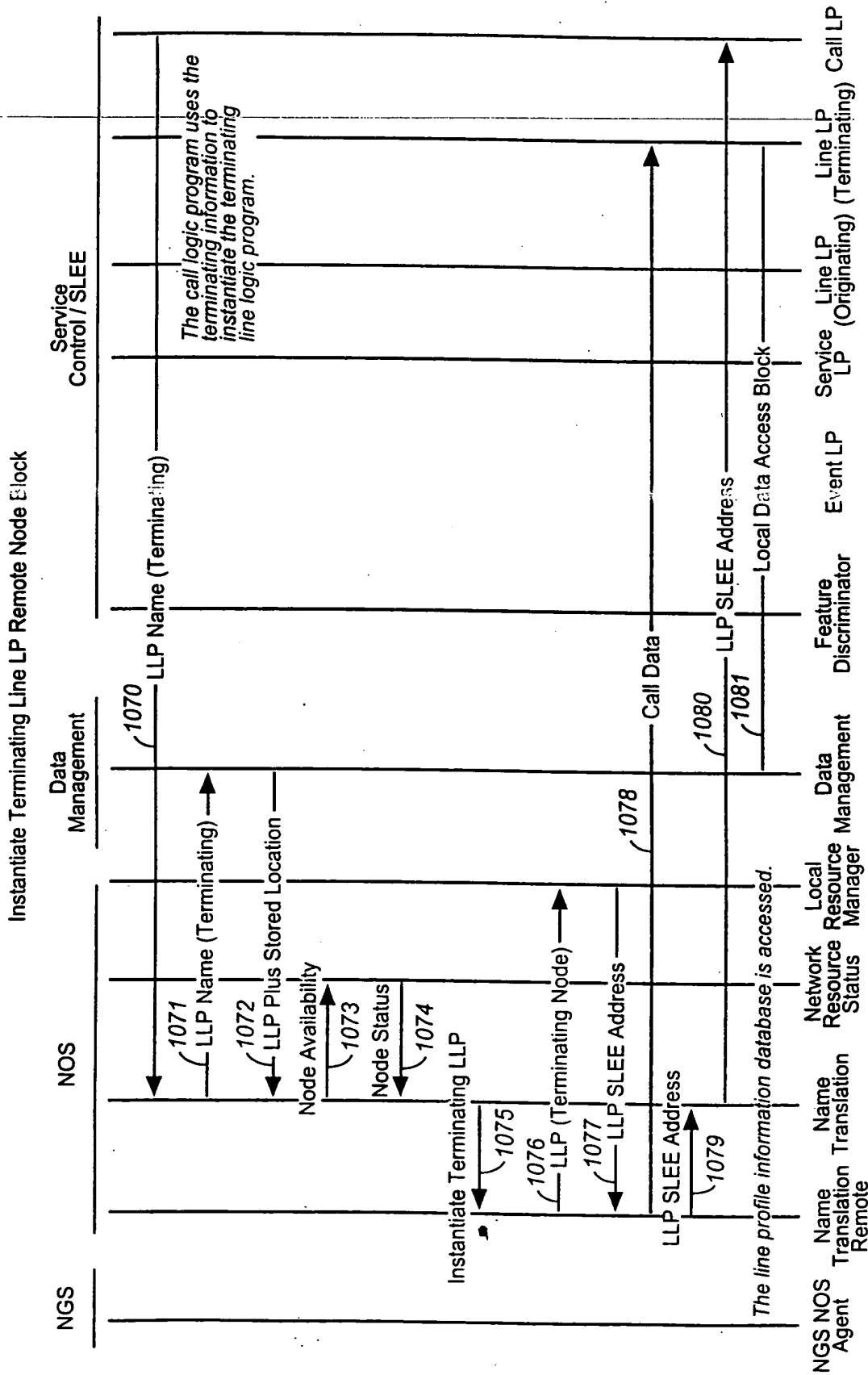
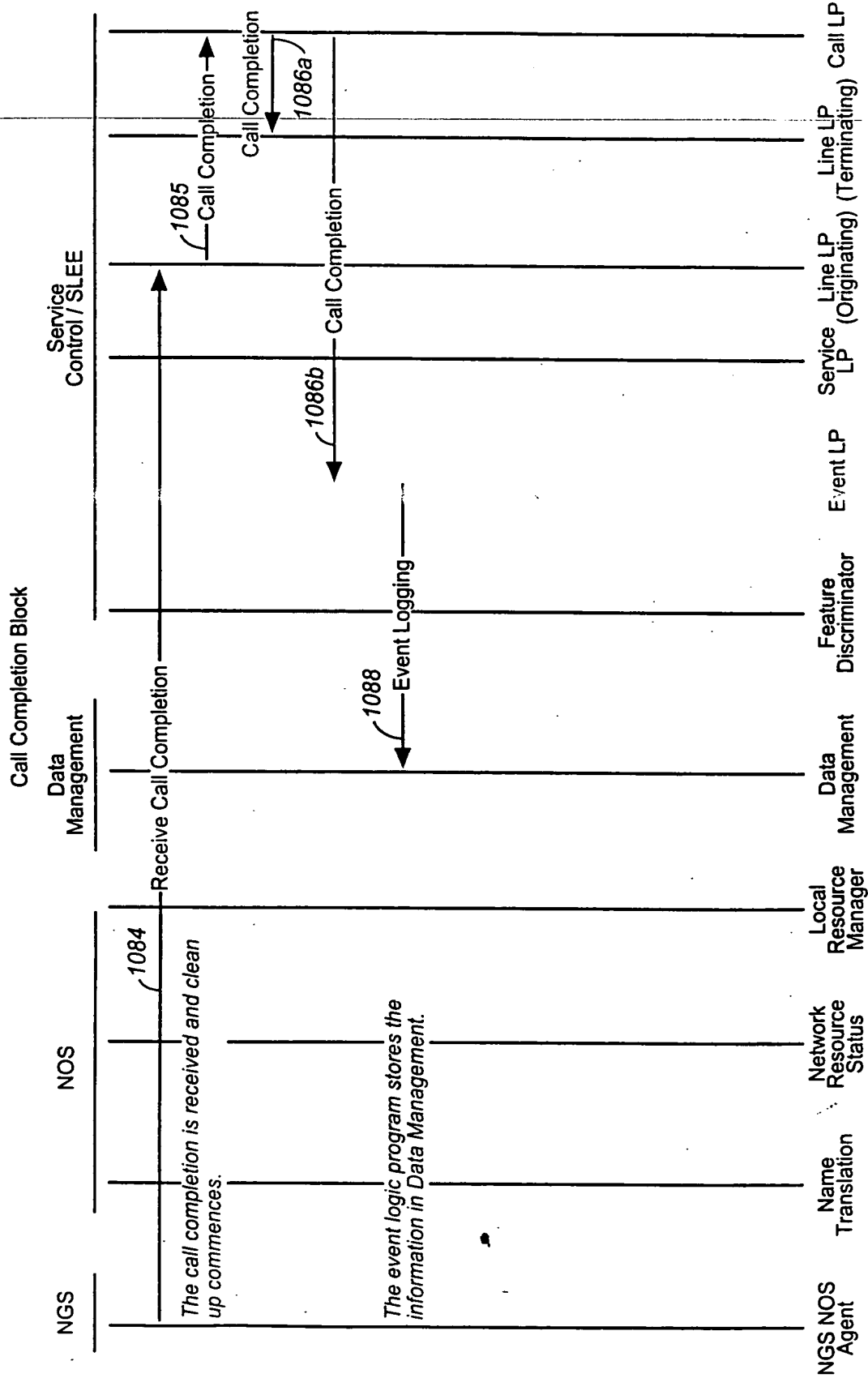


FIG. 18e



35/61

FIG. 18f

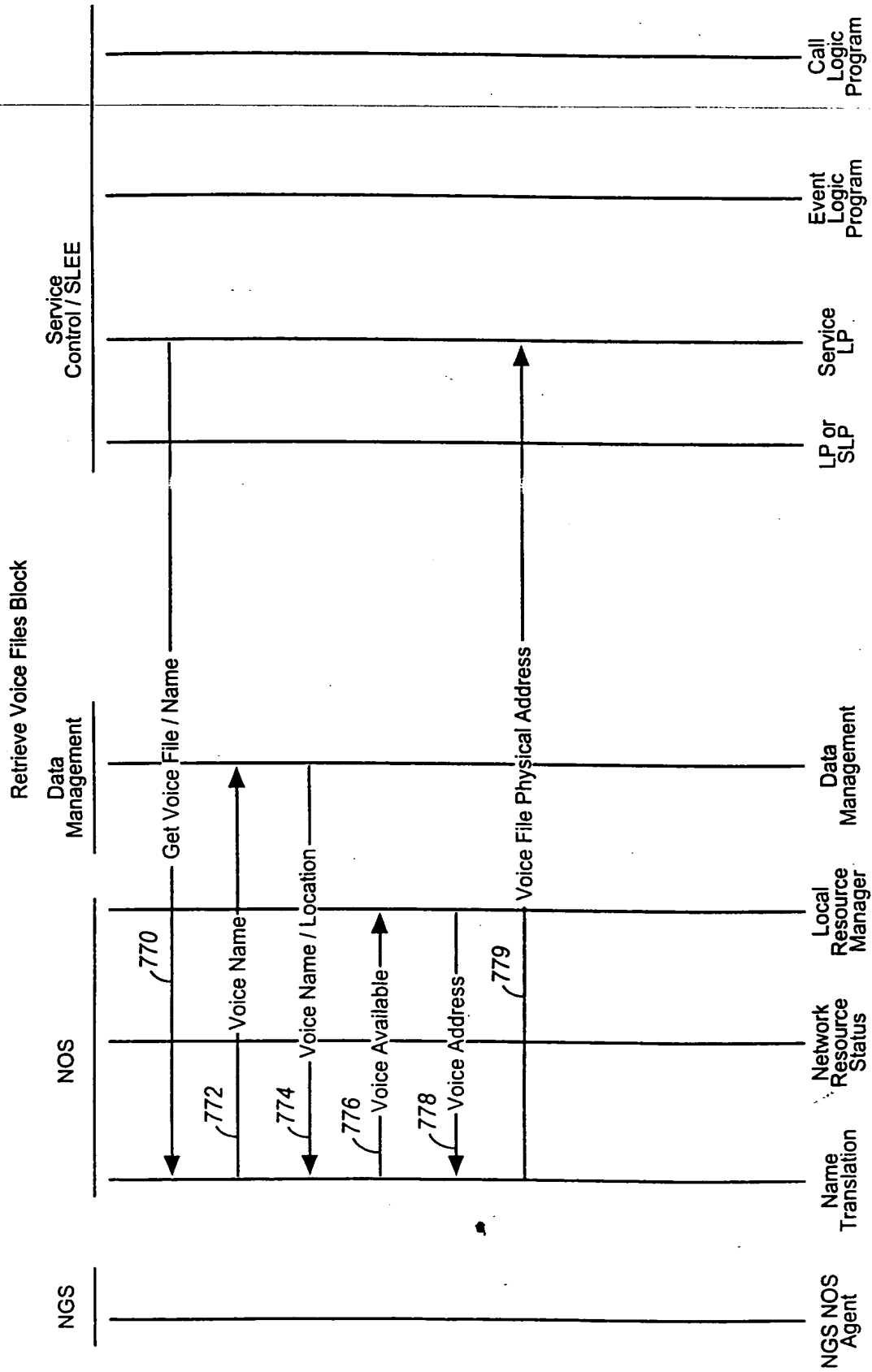
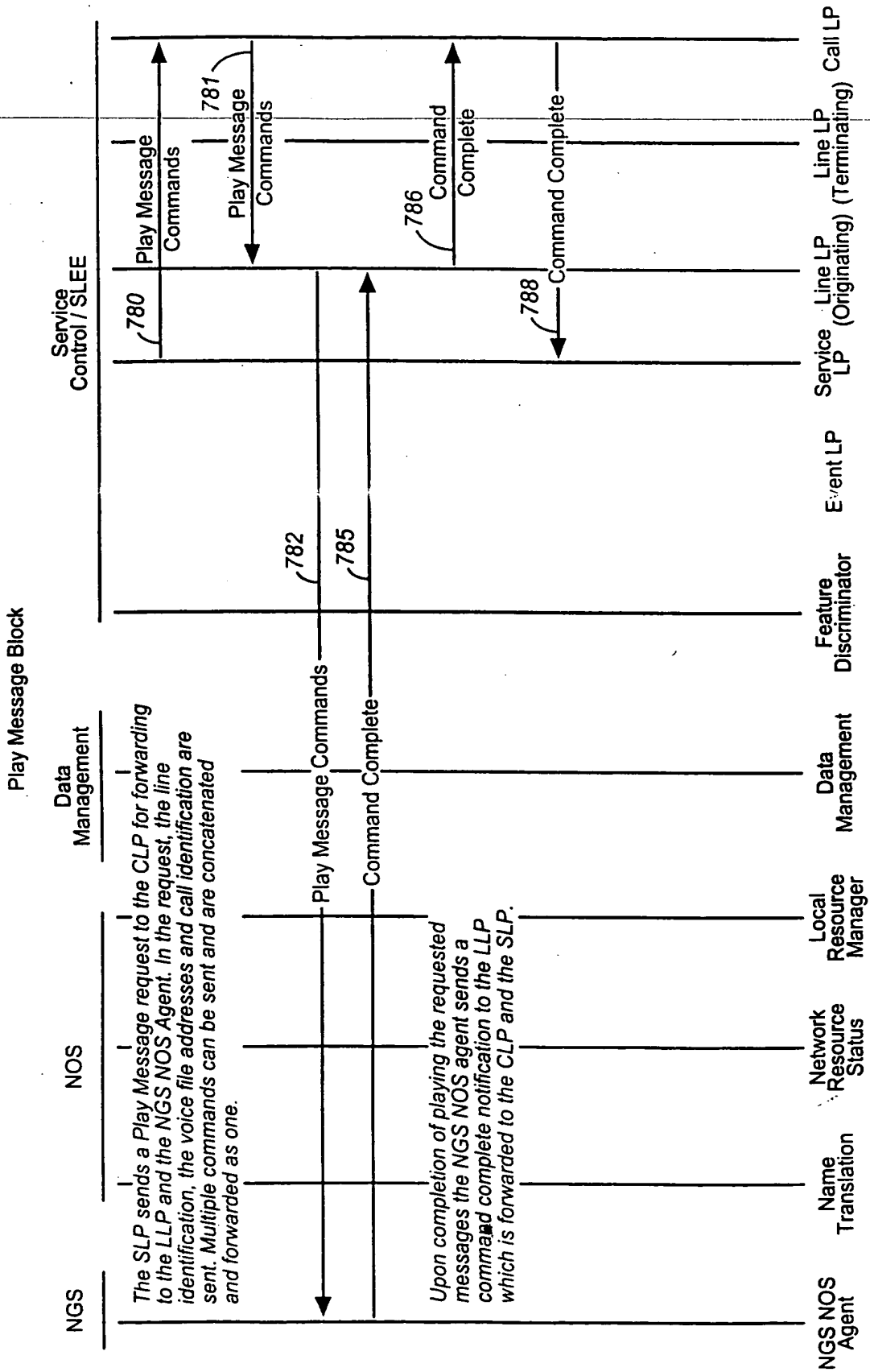


FIG. 18g



37/61

FIG. 18h

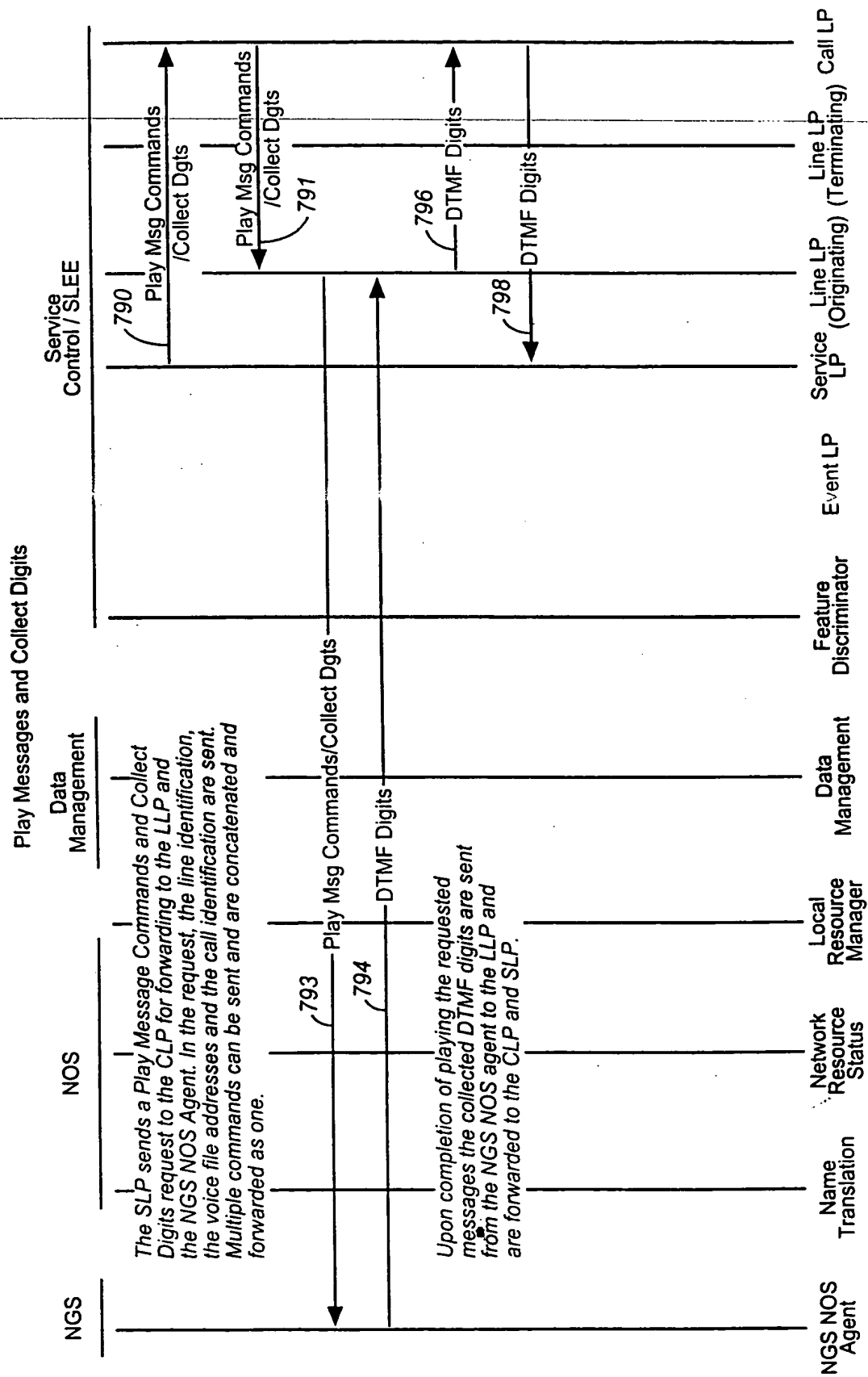
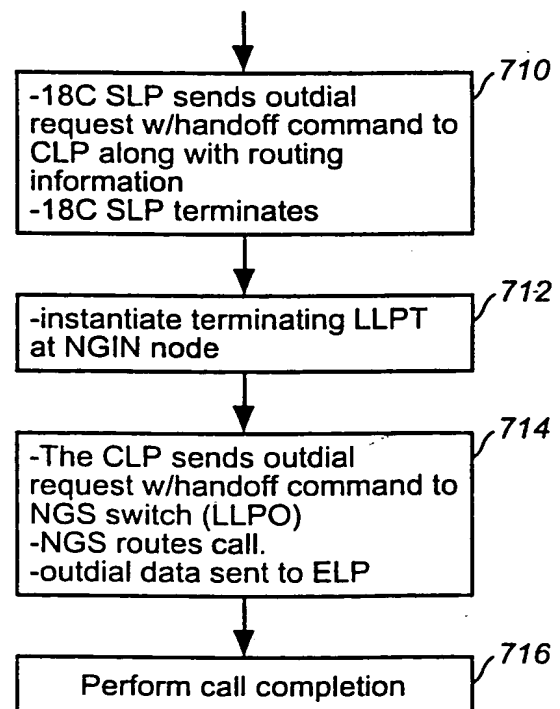
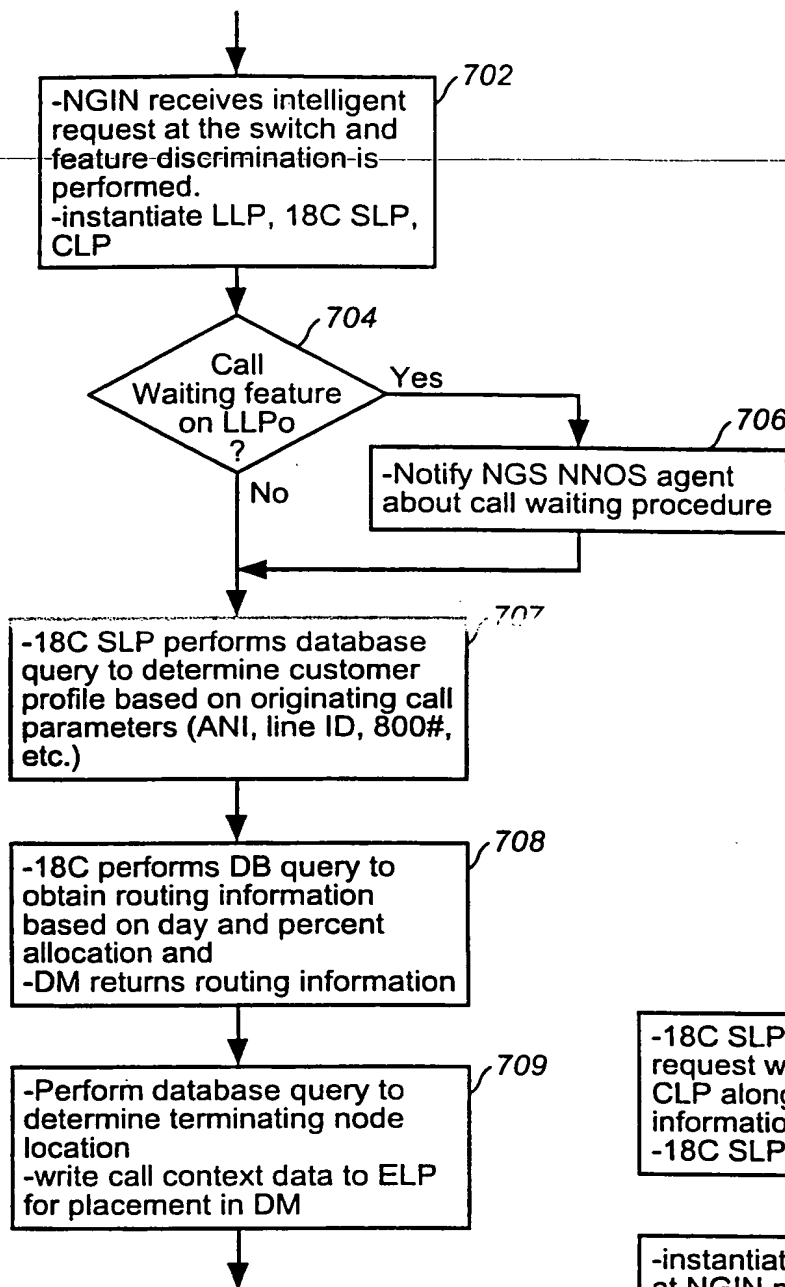


FIG. 18i

39/61



40/61

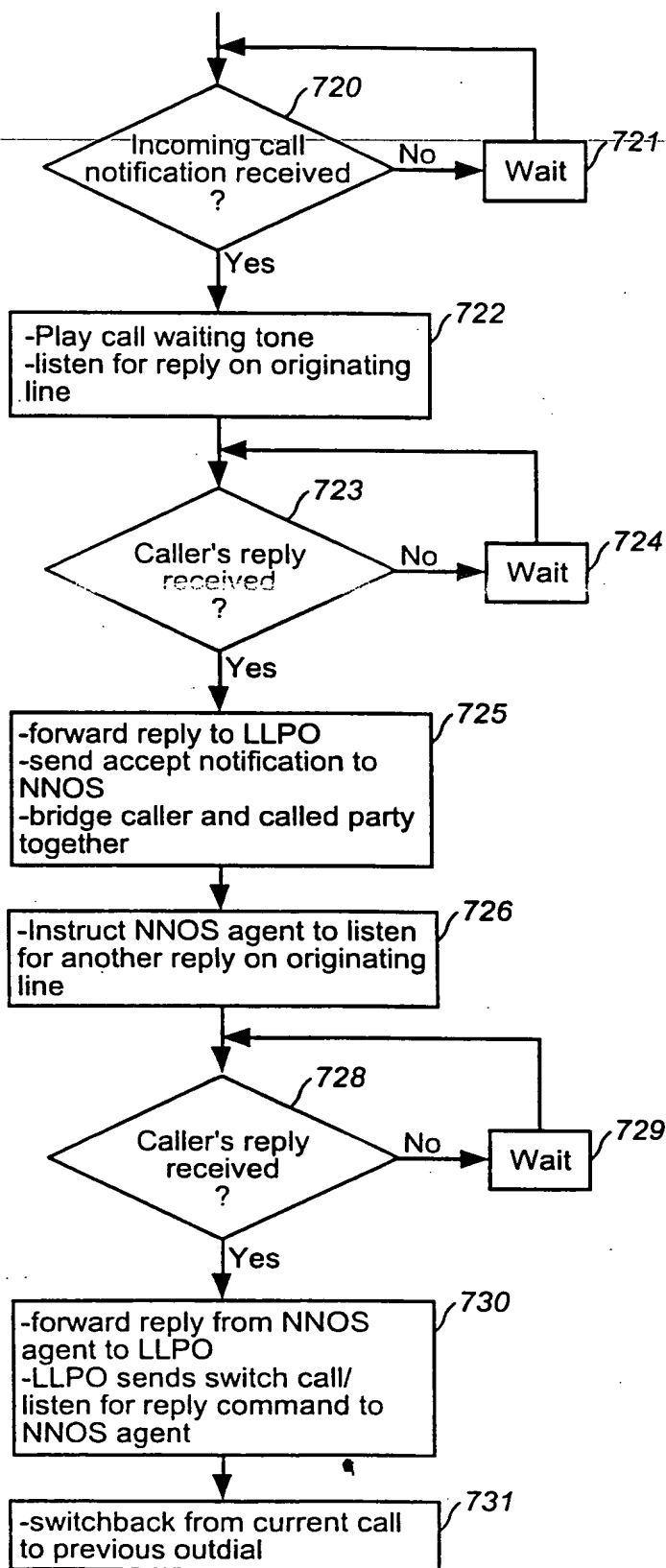


FIG. 19c

41/61

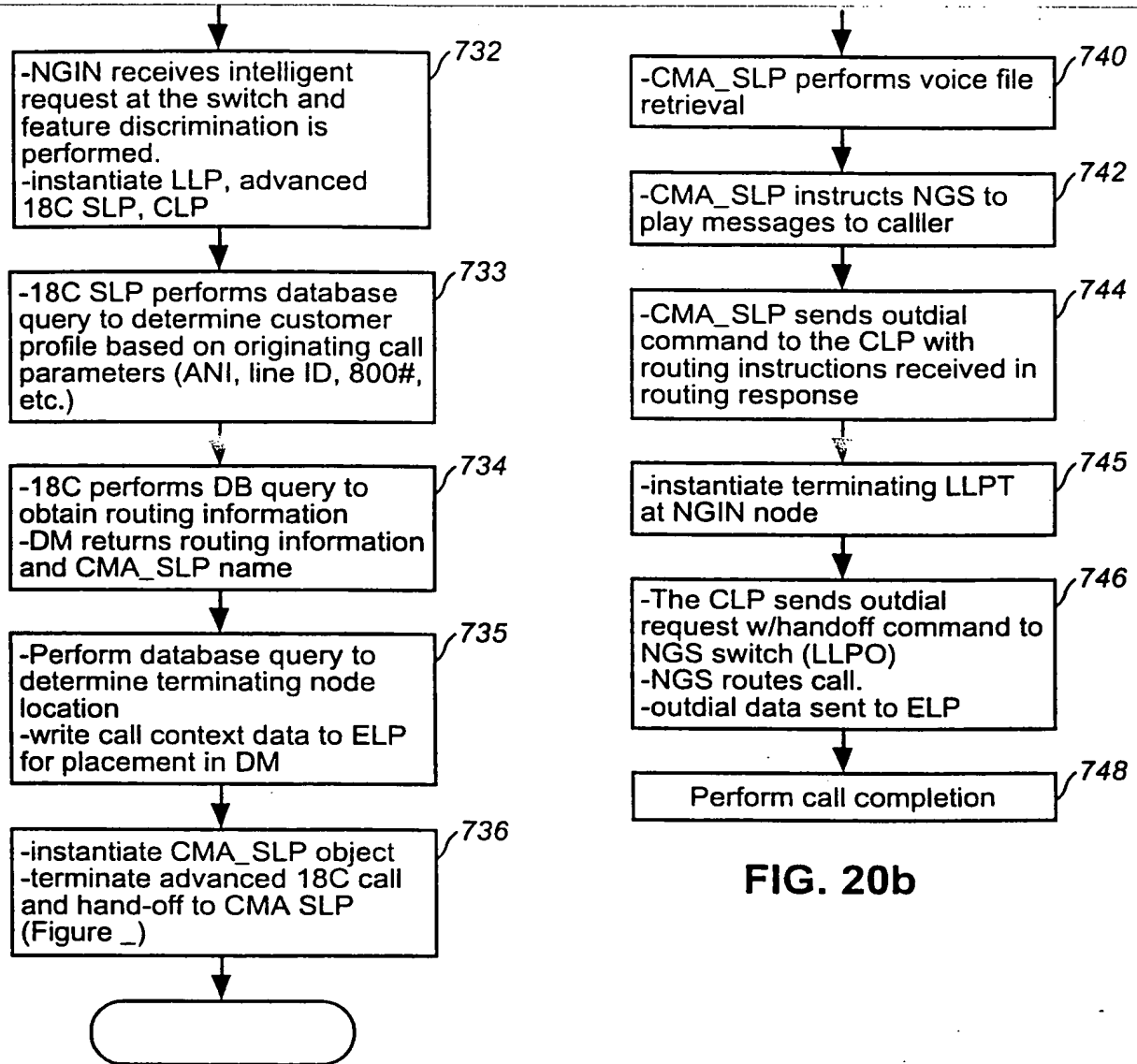
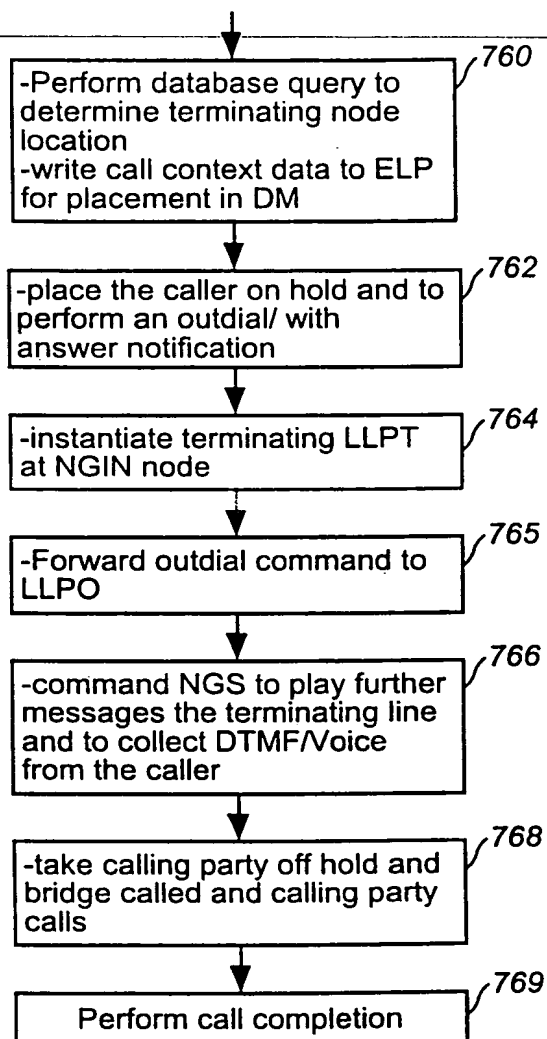
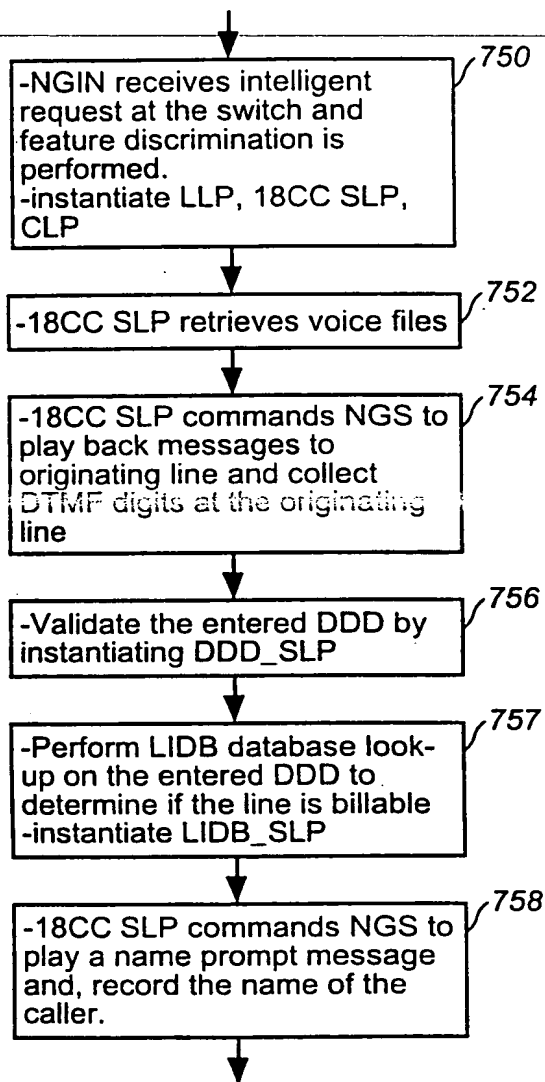


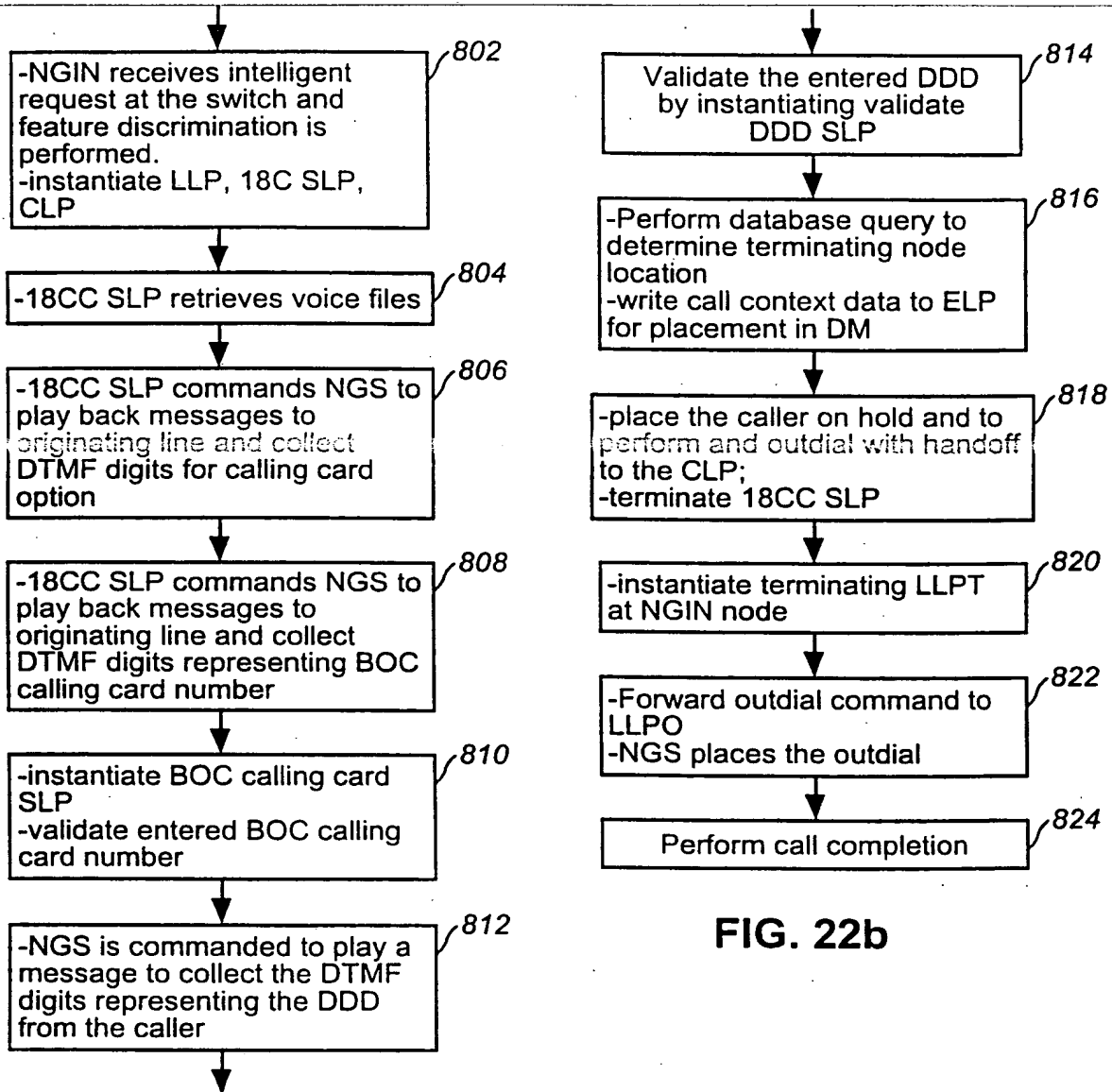
FIG. 20b

FIG. 20a

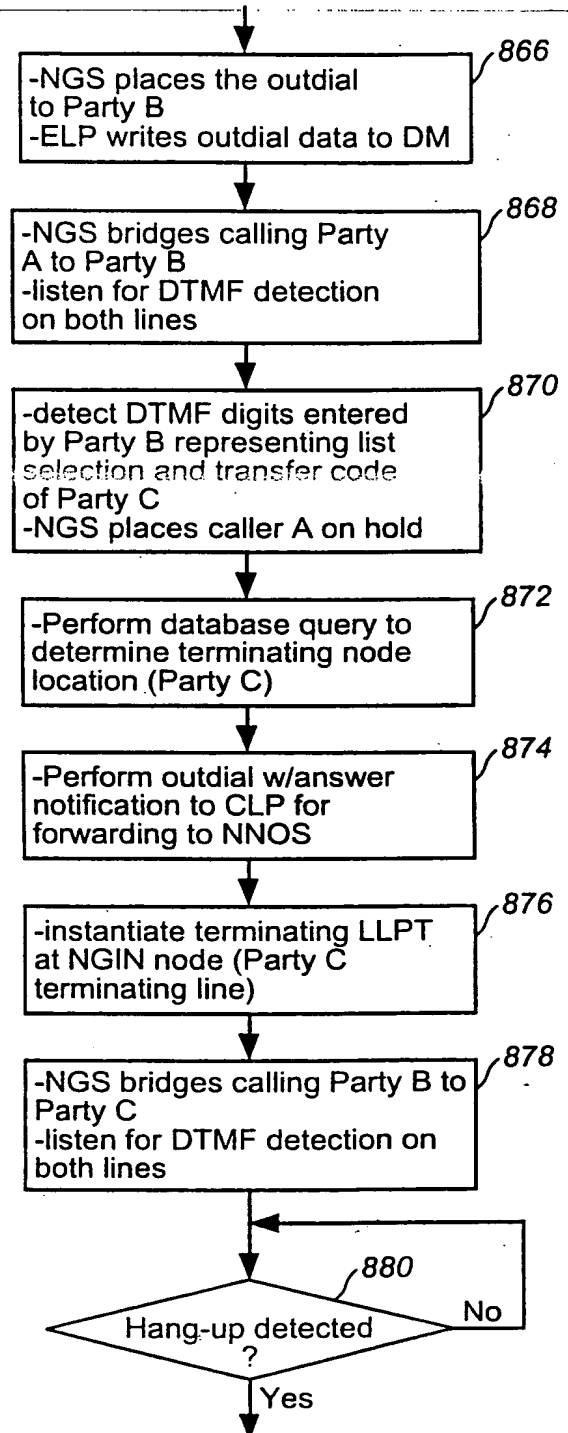
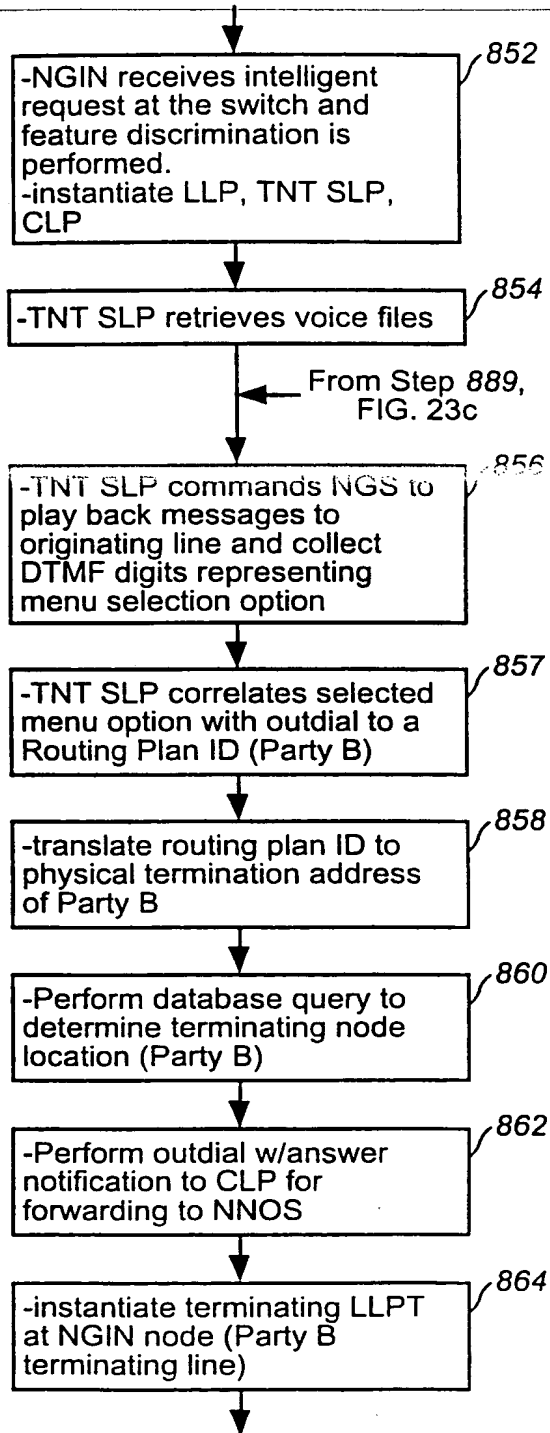
42/61



43/61



44/61



45/61

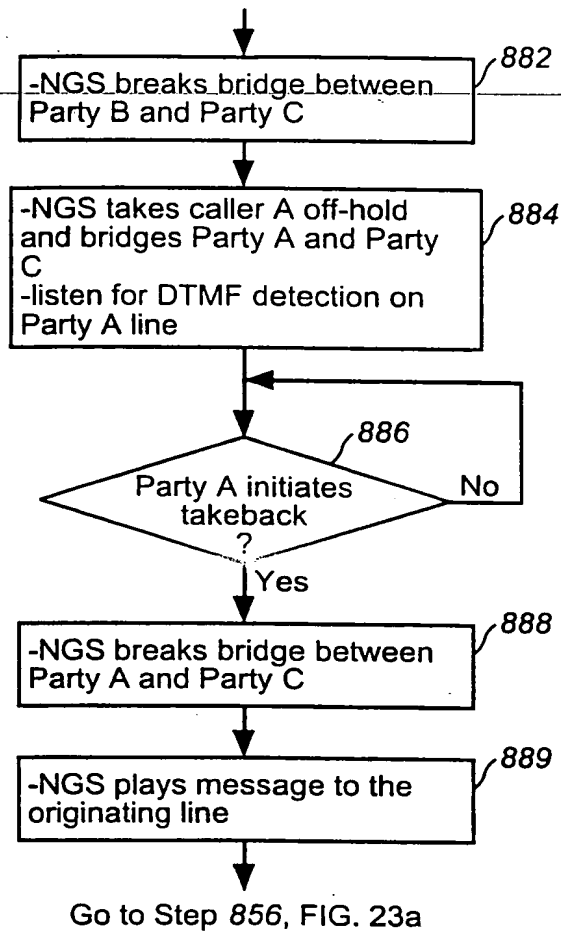
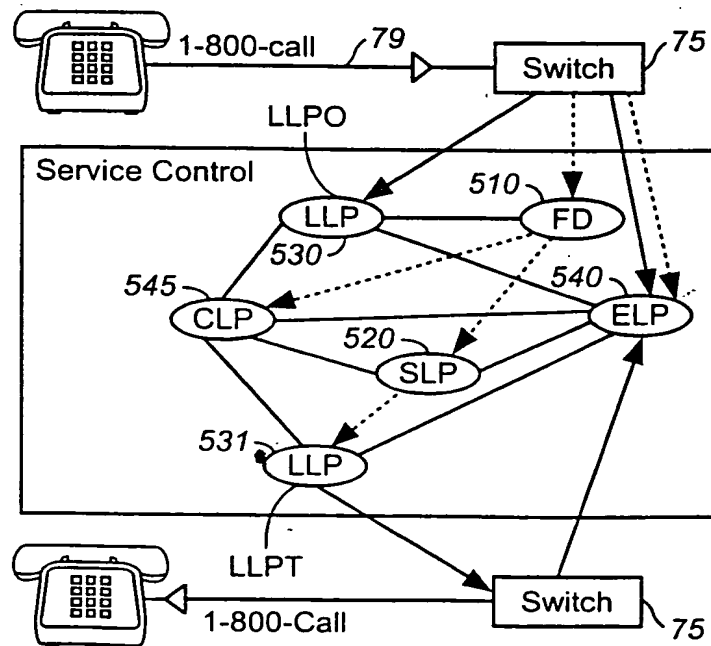
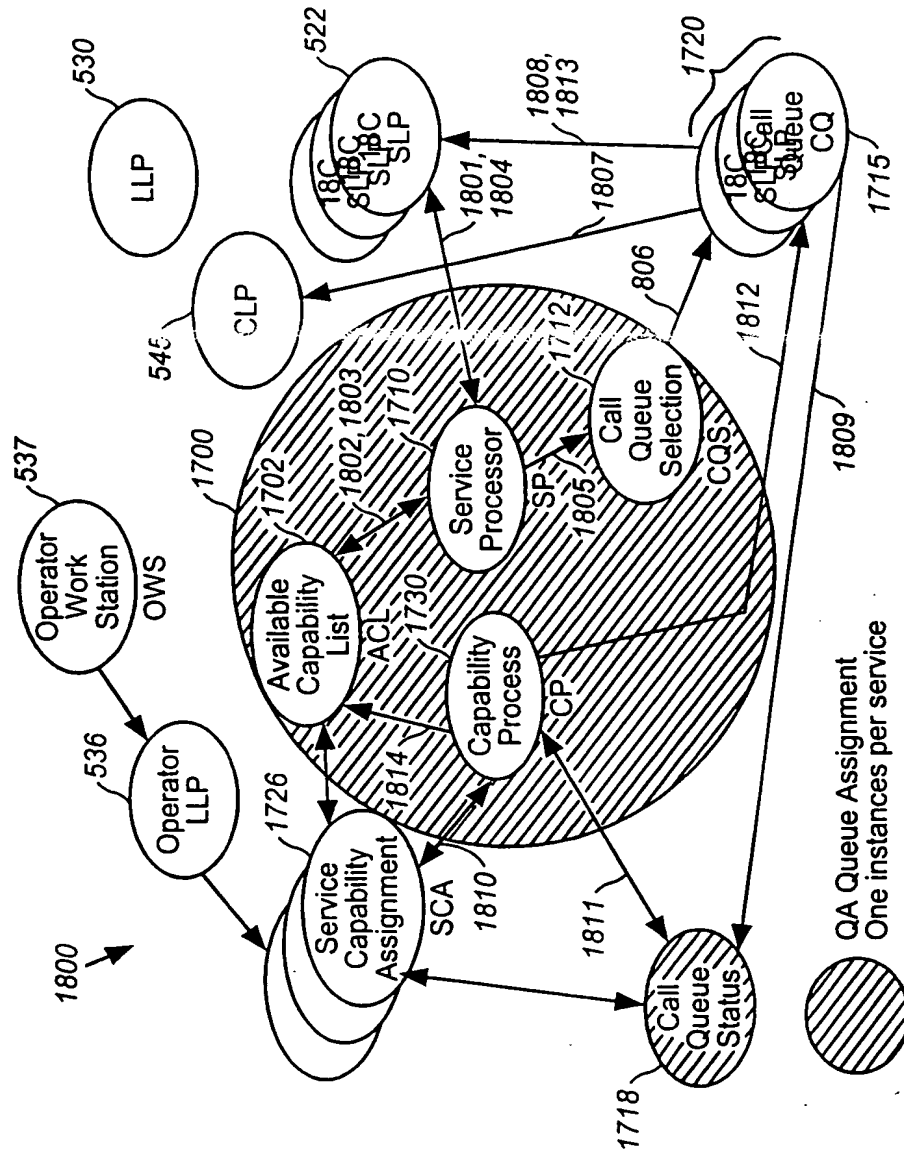


FIG. 24





47/61

1841	SLP Name: The 18C SLP sends the logical 18C QA SLP to NT for name translation.
1842	SLP Name: NT sends the logical 18C QA SLP Name to DM.
1843	SLP Name Plus Stored Locations: DM returns the actual 18C QA SLP name plus all the locations where it is stored.
1844	Node Selection Request: NT queries the NRS to determine the best node that is able to run the 18C QA SLP. This is done based on location and node status.
1845	Node Selection: NRS returns to NT the selected node.
1846	Instantiate Remote SLP: The NT of the local node requests the NT of the remote node to instantiate the 18C QA SLP.
1847	18C QA SLP (Remote Node): The NT on the remote node queries its LRM to determine if the 18C QA SLP is already instantiated on this node. If not, it instantiates the SLP.

FIG. 26a

1848	Query Data (Remote Node): The LRM of the remote node forwards the query data to the QA_SP. The query includes the return address of the 18C SLP and the associated CLP.
------	---

FIG. 26b

1849	LP Name: The 18C SLP sends the logical 18C QA_SP LP name to NT for name translation.
1850	LP Name: NT sends the logical 18C QA SP LP Name to DM.
1851	LP Name Plus Stored Locations: DM returns the actual 18C QA SLP name plus all the locations where it is stored.
1852	Node Selection Request: NT queries the NRS to determine the best node that is able to run the 18C QA SLP. This is done based on location and node status.
1853	Node Selection: NRS returns to NT the selected node.
1854	Instantiate Remote SLP: The NT of the local node requests the NT of the remote node to instantiate the 18C QA SLP.
1855	18C QA SLP (Remote Node): The NT on the remote node queries its LRM to determine if the 18C QA SLP is already instantiated on this node. If not, it instantiates the SLP.
1856	Query Data (Remote Node): The LRM of the remote node forwards the query data to the QA_ACL. The query includes the return address of the QA_SP.
1857	Query Answer: The QA_ACL processes the query and the result is returned to the QA_SP.

FIG. 26c

48/61

1858	LP Name: The QA_SP LP sends the logical QA_CQS LP name to NT for name translation.
1859	LP Name: NT sends the logical QA_CQS LP Name to DM.
1860	LP Name Plus Stored Locations: DM returns the actual QA_CQS LP name plus all the locations where it is stored. It is determined that this LP is stored locally and may be executed at this node.
1861	LP: The NT queries its LRM to determine if the QA_CQS LP is already instantiated at this node. If not, it instantiates the LP.
1862	LP SLEE Address: The LRM returns to NT the SLEE address where the QA_CQS LP is running.
1863	Data: The LRM forwards the data to the QA_CQS LP.

FIG. 26d

1864	LP Name: The QA_SP LP sends the logical CQ LP to NT for name translation.
1865	LP Name: NT sends the logical CQ LP Name to DM.
1866	SLP Name Plus Stored Locations: DM returns the actual CQ LP name plus all the locations where it is stored.
1867	Node Selection Request: NT queries the NRS to determine the best node that is able to run the CQ LP. This is done based on location and node status.
1868	Node Selection: NRS returns to NT the selected node.
1869	Instantiate Remote LP: The NT of the local node requests the NT of the remote node to instantiate the CQ LP.
1870	CQ LP (Remote Node): The NT on the remote node queries its LRM to determine if the CQ LP is already instantiated on this node. If not, it instantiates the LP.
1871	Data (Remote Node): The LRM of the remote node forwards the data to the CQ.

FIG. 26e

1881	LP Name: The SCA LP sends the logical QA_CP LP name to NT for name translation.
1882	LP Name: NT sends the logical QA_CP LP Name to DM.
1883	LP Name Plus Stored Locations: DM returns the actual QA_CP LP name plus all the locations where it is stored.
1884	Node Selection Request: NT queries the NRS to determine the best node that is able to run the QA_CP LP. This is done based on location and node status.
1885	Node Selection: NRS returns to NT the selected node.
1886	Instantiate Remote SLP: The NT of the local node requests the NT of the remote node to instantiate the QA_CP LP.
1887	QA_CP LP (Remote Node): The NT on the remote node queries its LRM to determine if the QA_CP LP is already instantiated on this node. If not, it instantiates the SLP.
1888	Data (Remote Node): The LRM of the remote node forwards the data to the QA_CP LP. This data will contain the logical name of the operator termination location.

FIG. 26f

49/61

1889	LP Name: The CP LP sends the logical Call Queue Status LP name to NT for name translation.
1890	LP Name: NT sends the logical Call Queue Status LP Name to DM.
1891	LP Name Plus Stored Locations: DM returns the actual Call Queue Status LP name plus all the locations where it is stored.
1892	Node Selection Request: NT queries the NRS to determine the best node that is able to run the Call Queue Status LP. This is done based on location and node status.
1893	Node Selection: NRS returns to NT the selected node.
1894	Instantiate Remote SLP: The NT of the local node requests the NT of the remote node to instantiate the Call Queue Status LP.
1895	Call Queue Status LP (Remote Node): The NT on the remote node queries its LRM to determine if the Call Queue Status LP is already instantiated on this node. If not, it instantiates the SLP.
1896	Query (Remote Node): The LRM of the remote node forwards the query to the Call Queue Status LP.
1897	Response (Remote Node): The LRM of the remote node forwards the response to the CP. The response will contain the address of the CQ and the SLP which is waiting on the capability.

FIG. 26g

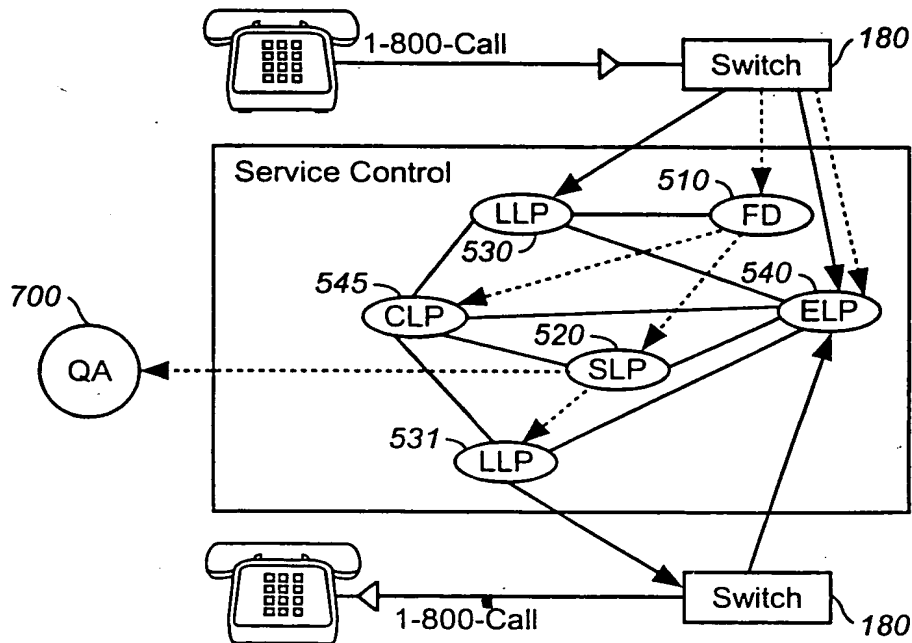


FIG. 29

50/61

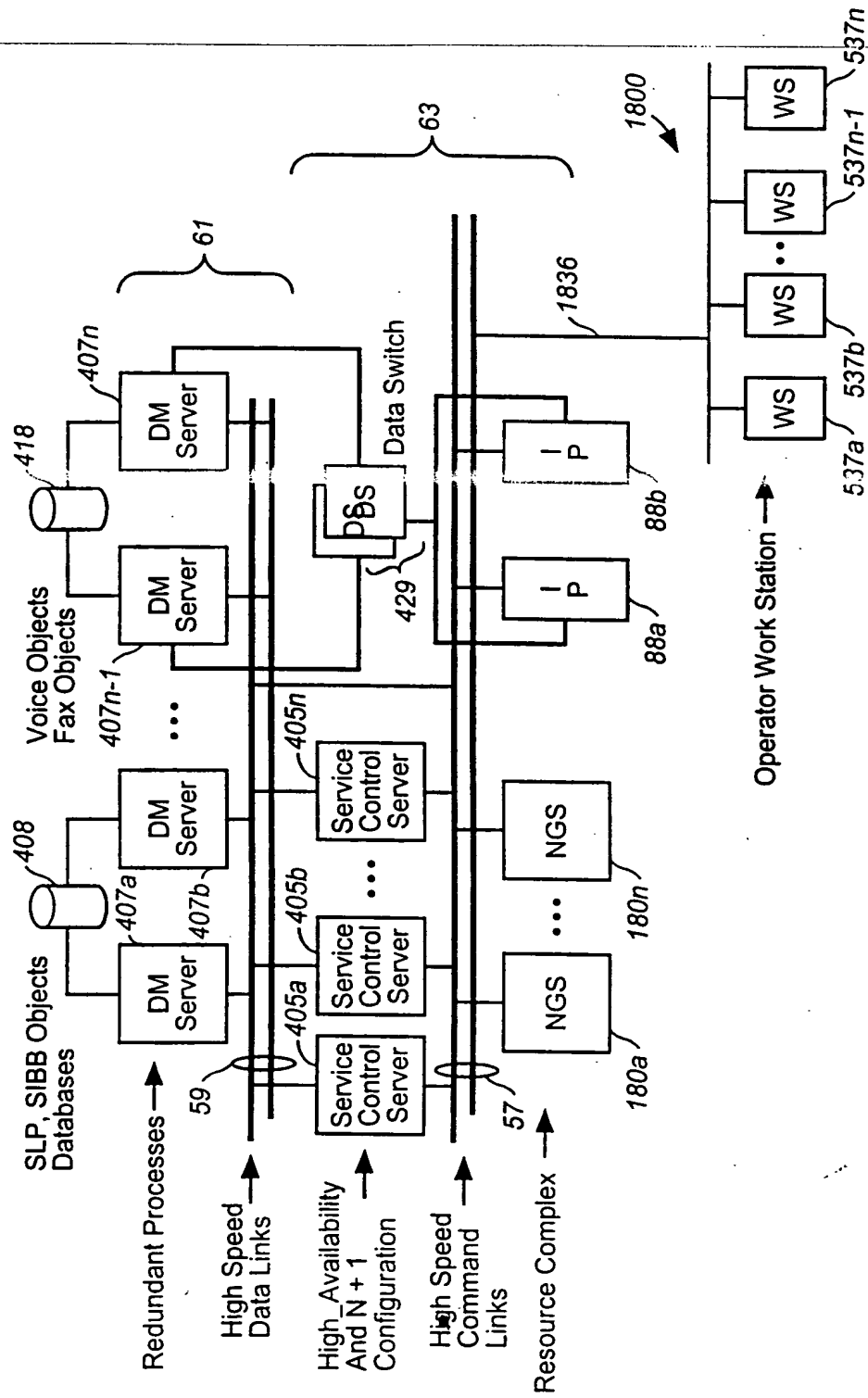


FIG. 27a

51/61

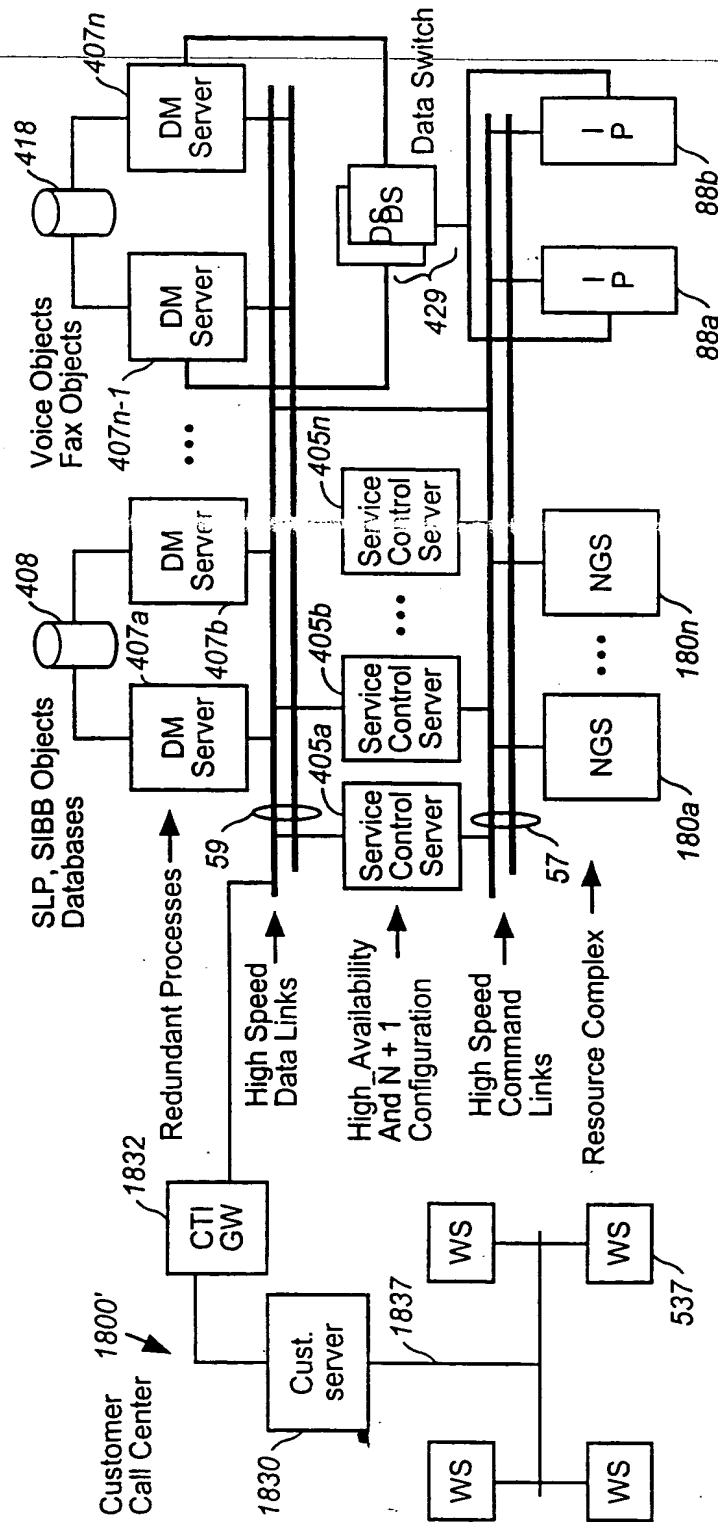


FIG. 27b

52/61

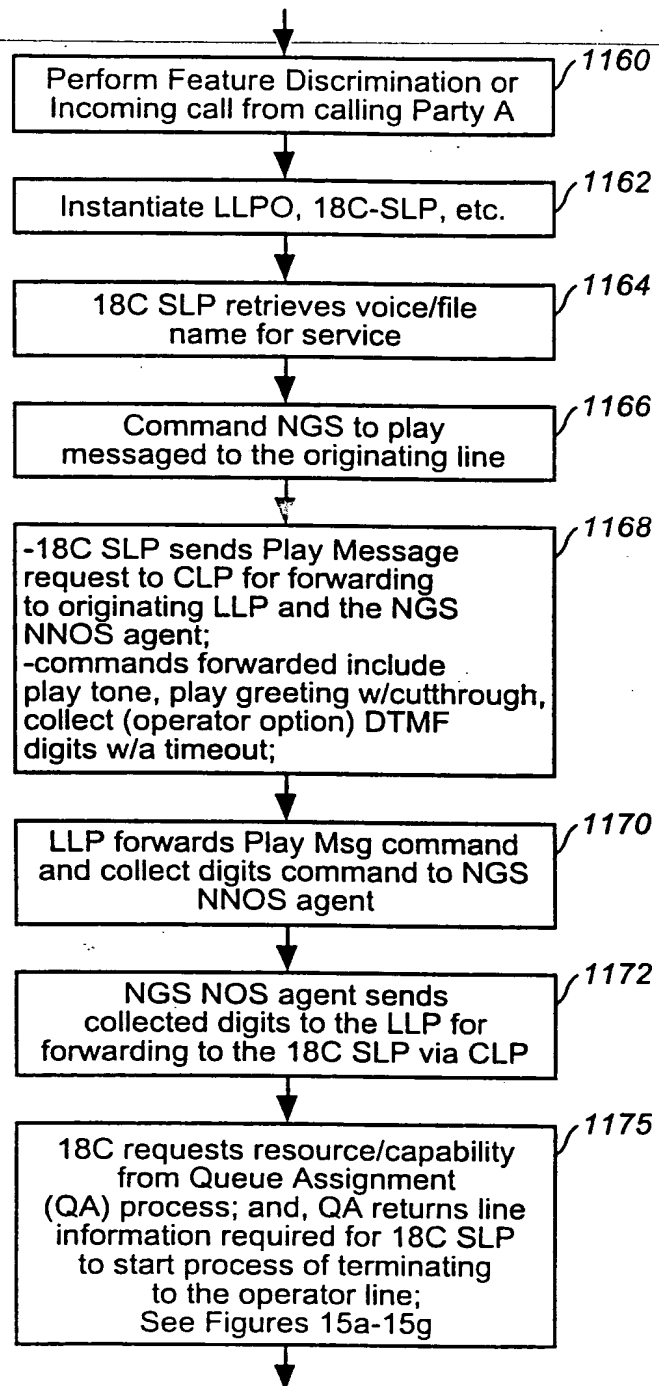


FIG. 28a

53/61

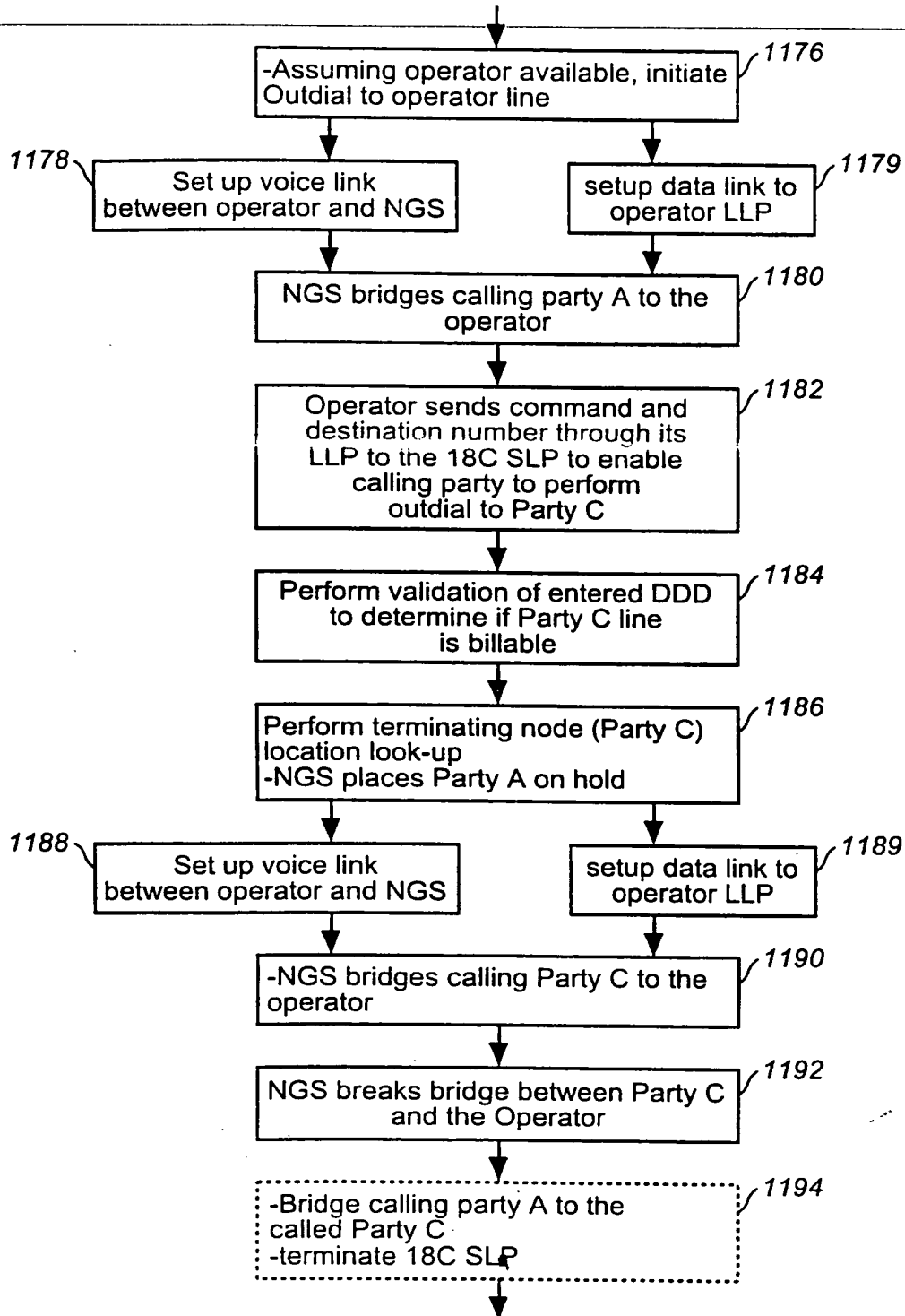


FIG. 28b

54/61

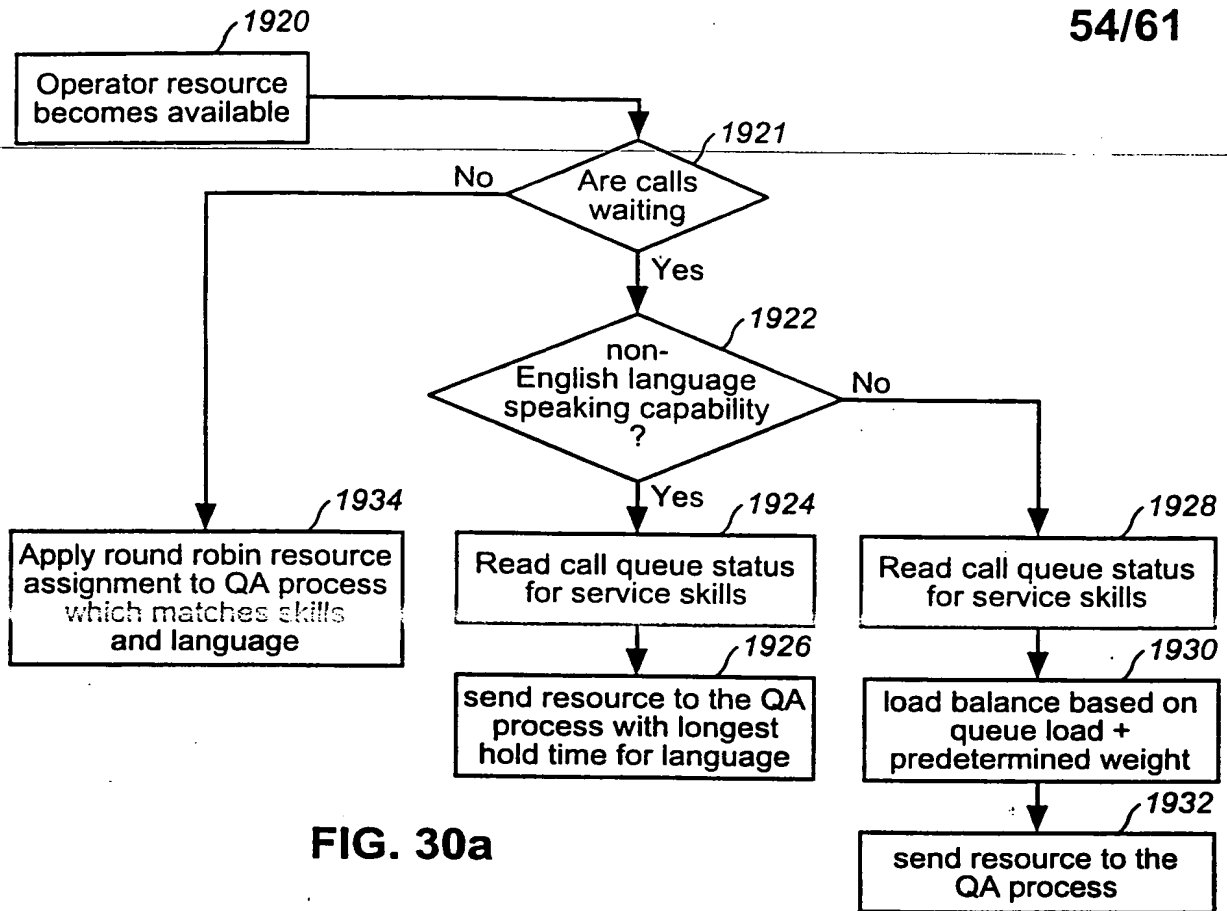


FIG. 30a

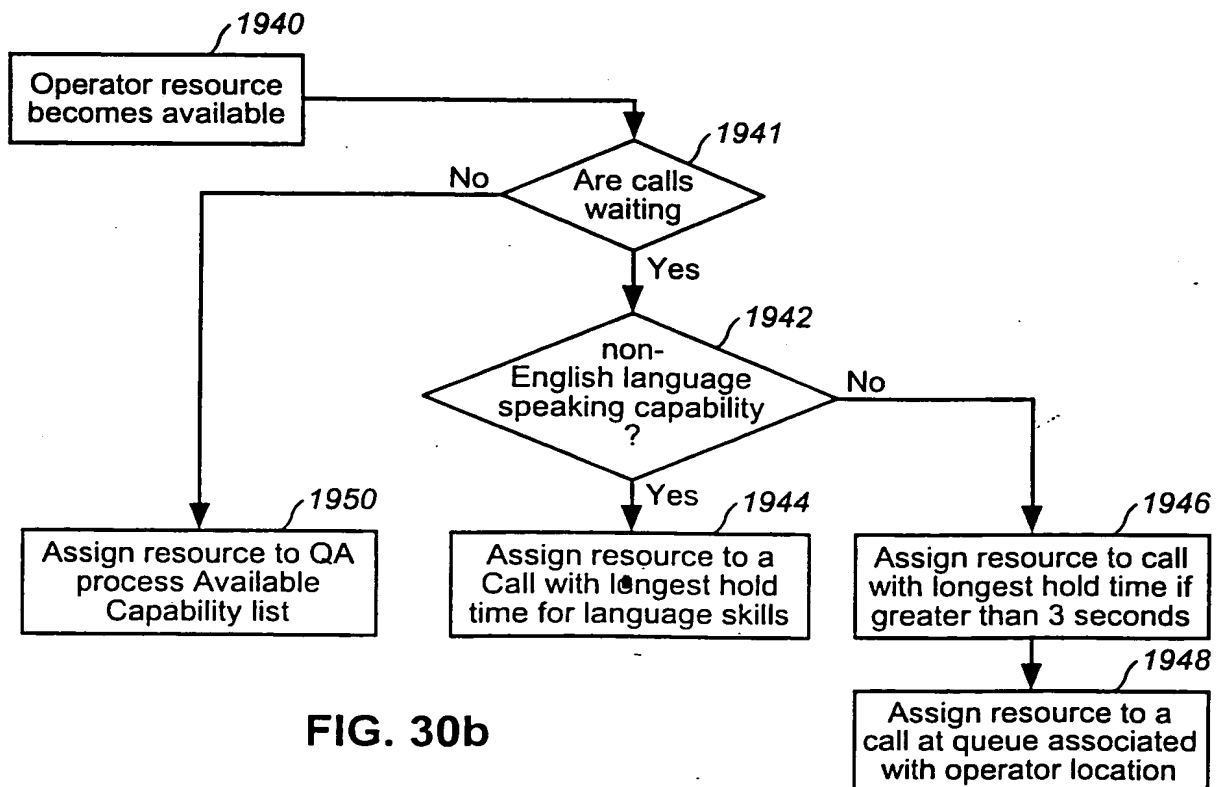


FIG. 30b

55/61

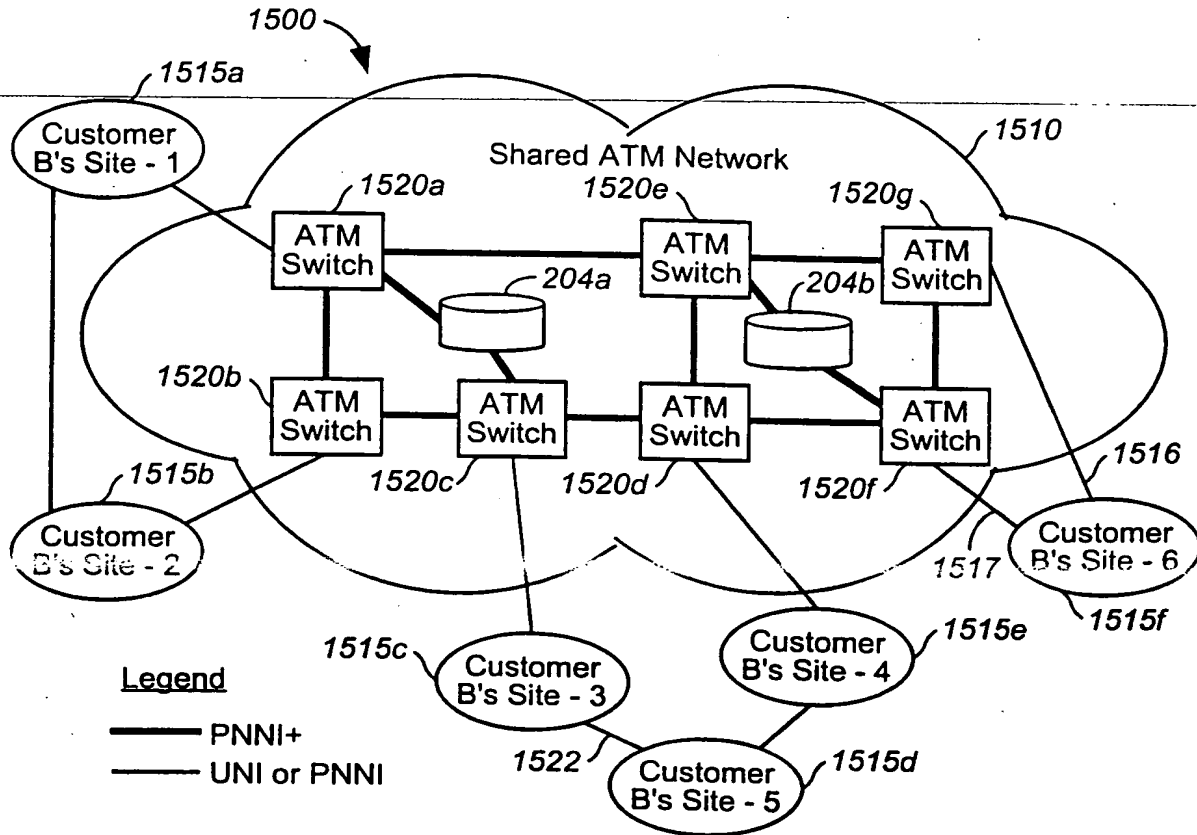


FIG. 31a

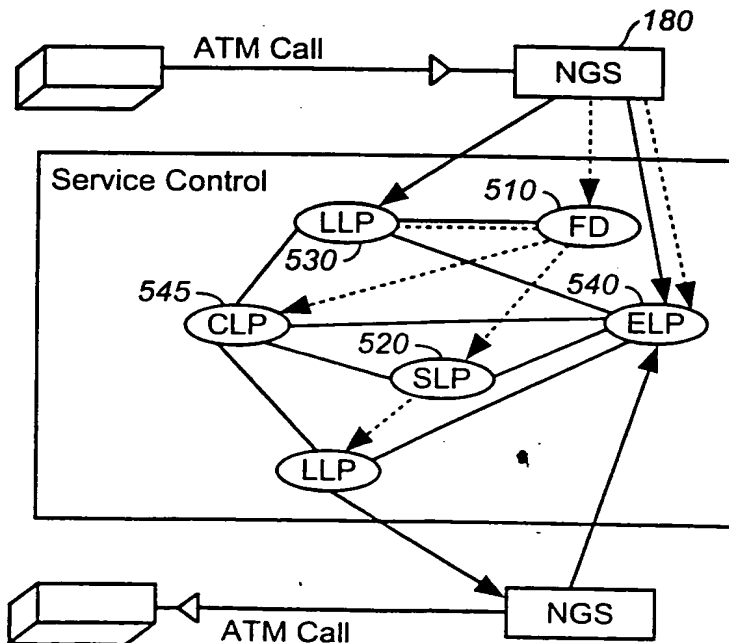


FIG. 31b

56/61

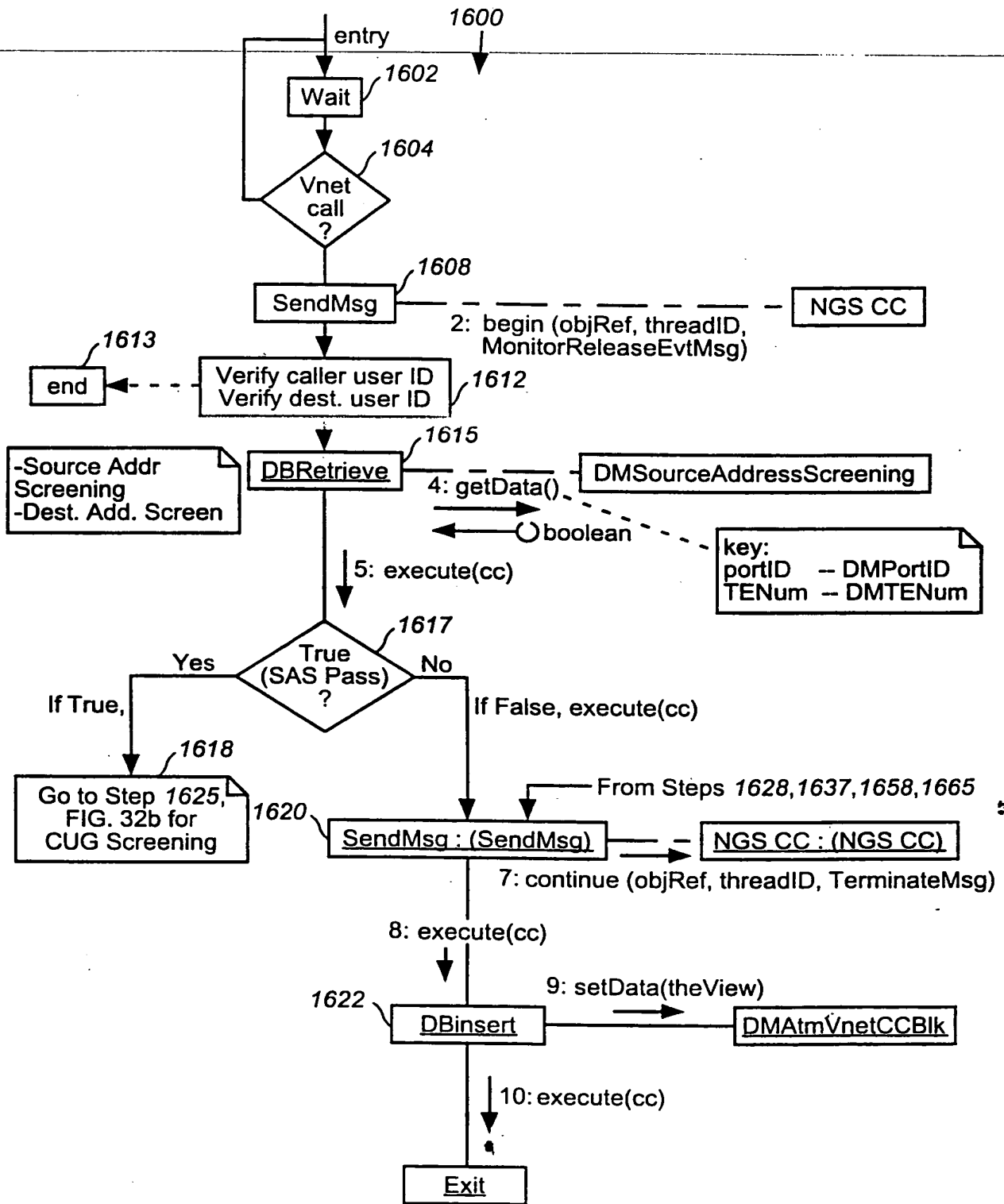


FIG. 32a

```

graph TD
    Start([Continue From Step 1618, FIG. 32a  
CUG Screening]) --> DBRetrieve[DBRetrieve 1625]
    DBRetrieve -- "1: getData()" --> DMCUGScreening[DMCUGScreening]
    DMCUGScreening -- "true/false" --> DBRetrieve
    DBRetrieve -- "2: execute(cc)" --> Decision{True (CUGS Pass) ? 1628}
    Decision -- No --> Step1620[Go to Step 1620, FIG. 32a]
    Decision -- Yes --> Step1630[Go to Step 1630, FIG. 32c  
TOY Routing]

```

key:
calling group -- DMCUG
called group -- DMCUG

1: getData()

2: execute(cc)

1625 DBRetrieve

DMCUGScreening

1628 True (CUGS Pass) ?

No

Yes

if False,

if True,

1629

Go to Step 1620, FIG. 32a

Go to Step 1630, FIG. 32c
TOY Routing

FIG. 32b

```

graph TD
    Start([Continue From Step 1629, FIG. 32b  
TOY Routing]) --> GetTime[GetCurrentTime 1630]
    GetTime -- "1: execute(cc)" --> DBRetrieve[DBRetrieve 1633]
    DBRetrieve -- "2: getData()" --> DMCUG[DMCUG / Null]
    DMCUG -- "Called Party's TOY route choice" --> DBRetrieve
    DBRetrieve -- "3: execute(cc)" --> Decision{null returned? 1635}
    Decision -- Yes --> Step1640[Go to Step 1640, FIG. 32d  
TOD Routing 1637]
    Decision -- No --> Step1650[Go to Step 1650, FIG. 32e  
Route Plan 1638]

```

FIG. 32c

58/61

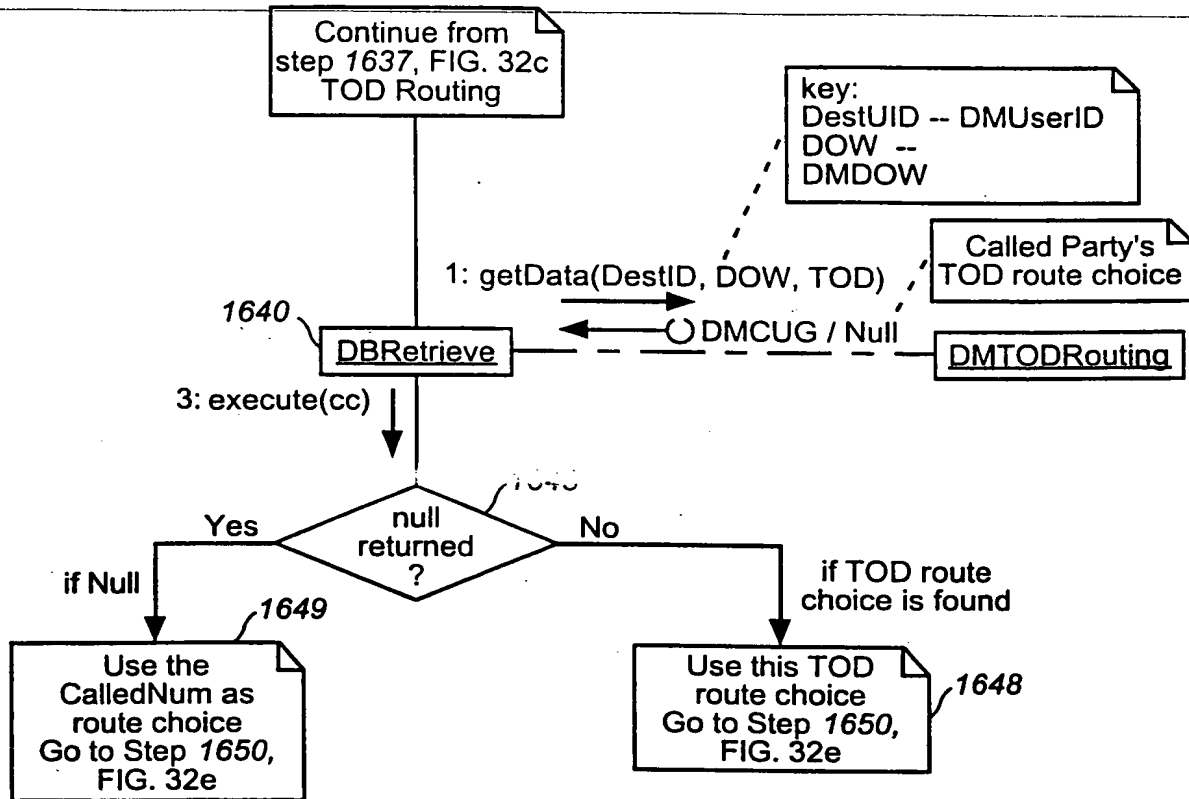


FIG. 32d

59/61

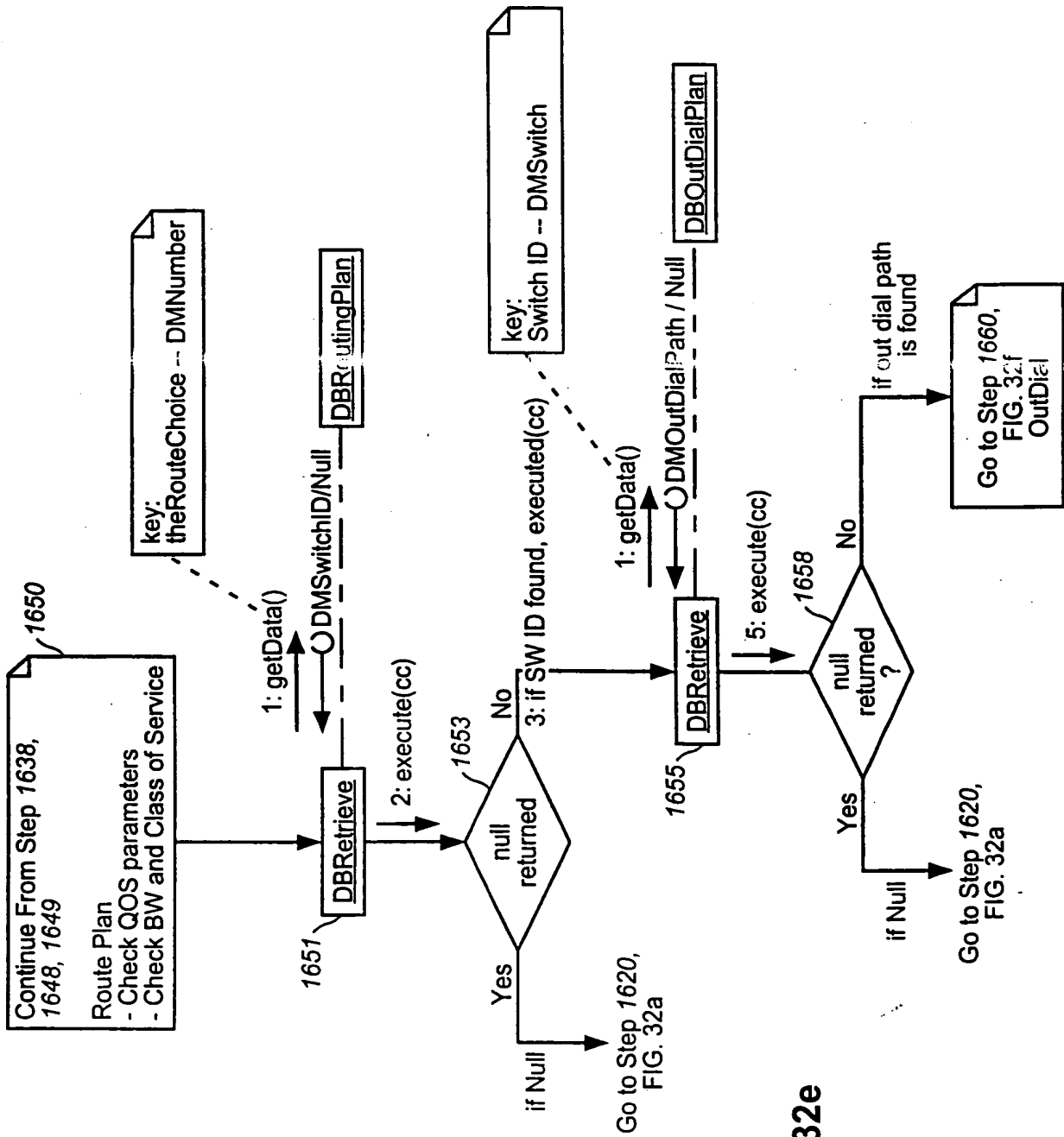
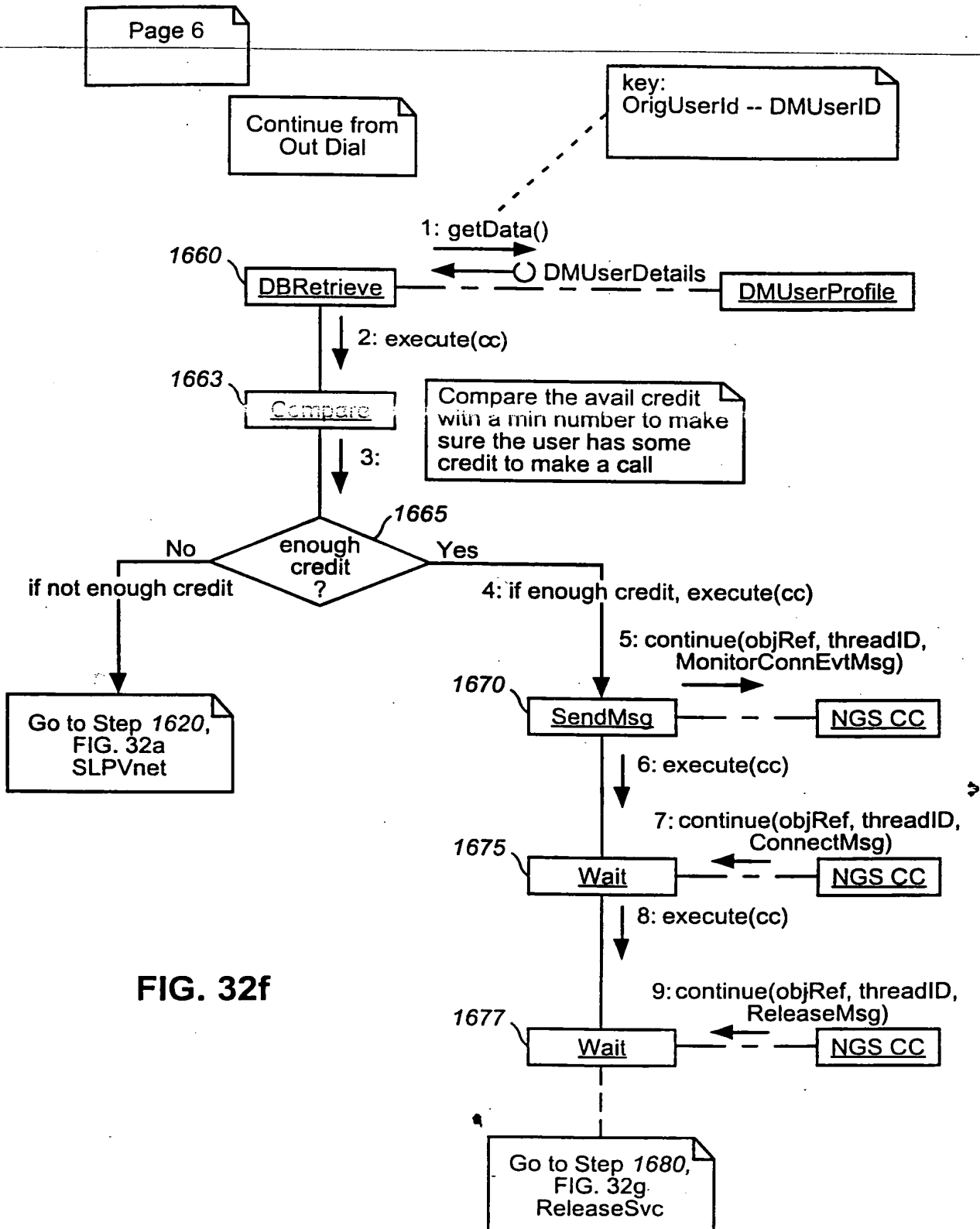


FIG. 32e

60/61



61/61

Page 7

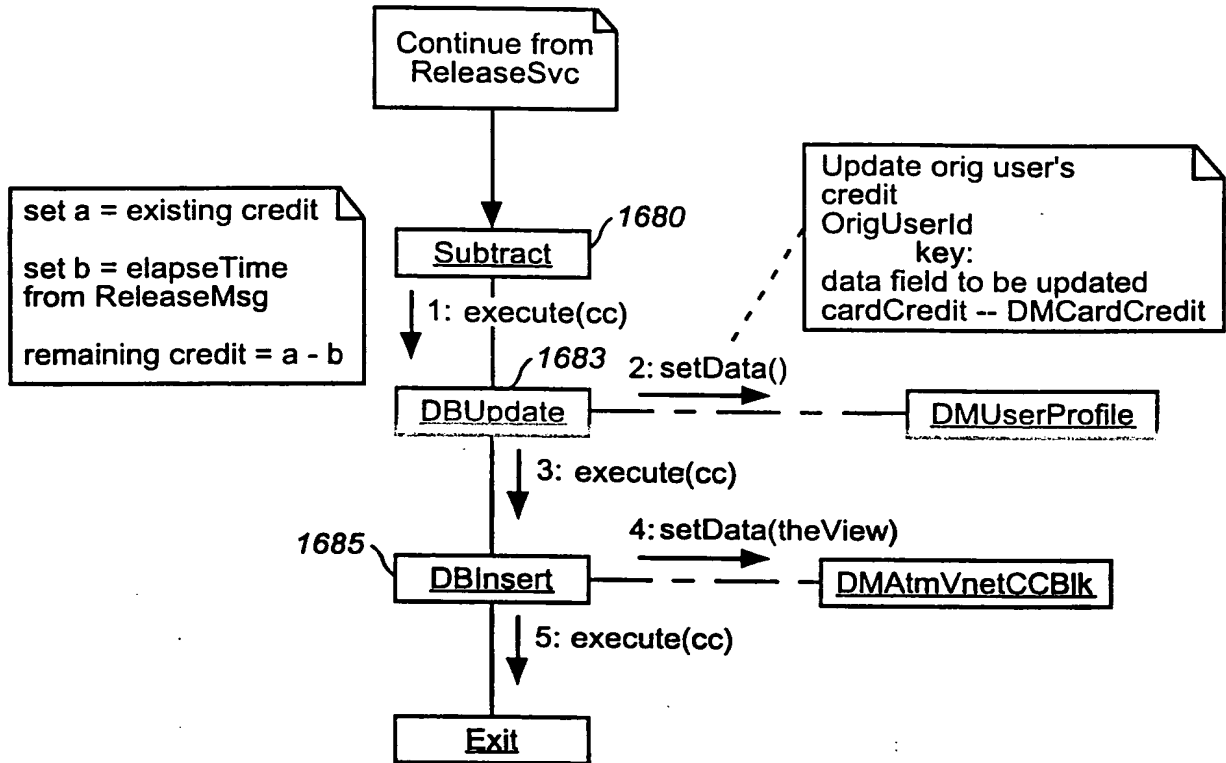


FIG. 32g